Red Hat Enterprise Linux 6 Sicherheitshandbuch

Anleitung zur Sicherung von Red Hat Enterprise Linux
Ausgabe 1.5

Red Hat Inc.

Red Hat Enterprise Linux 6 Sicherheitshandbuch

Anleitung zur Sicherung von Red Hat Enterprise Linux
Ausgabe 1.5

Red Hat Inc.

Rechtlicher Hinweis

Zusammenfassung

Dieses Handbuch hilft Benutzern und Administratoren beim Verständnis der notwendigen Prozesse und Verfahrensweisen zur Sicherung von Arbeitsplatzrechnern und Servern gegen Angriffe von innen und außen, Ausnutzen von Sicherheitslücken und böswilligen Aktivitäten. Das Handbuch beschreibt die Planung und die nötigen Werkzeuge zur Einrichtung einer sicheren Rechenumgebung für Rechenzentren, Arbeitsplatzrecher oder Heimcomputer. Dabei wird das Hauptaugenmerk zwar auf Red Hat Enterprise Linux gelegt, die Konzepte und Techniken sind jedoch auf alle Linux-Systeme übertragbar. Mit ausreichenden administrativen Kenntnissen, Wachsamkeit und den richtigen Werkzeugen können Linux-Systeme voll funktionsfähig gehalten werden und gleichzeitig vor den meisten üblichen Angriffen geschützt werden.

Inhaltsverzeichnis

Vorwort

1. Dokumentkonventionen

Dieses Handbuch verwendet mehrere Konventionen, um bestimmte Wörter und Sätze hervorzuheben und Aufmerksamkeit auf bestimmte Informationen zu lenken.

In PDF- und Papierausgaben verwendet dieses Handbuch Schriftbilder des Liberation-Fonts-Sets. Das Liberation-Fonts-Set wird auch für HTML-Ausgaben verwendet, falls es auf Ihrem System installiert ist. Falls nicht, werden alternative, aber äquivalente Schriftbilder angezeigt. Beachten Sie: Red Hat Enterprise Linux 5 und die nachfolgende Versionen beinhalten das Liberation-Fonts-Set standardmäßig.

1.1. Typografische Konventionen

Es werden vier typografische Konventionen verwendet, um die Aufmerksamkeit auf bestimmte Wörter und Sätze zu lenken. Diese Konventionen und die Umstände, unter denen sie auftreten, sind folgende:

`Nichtproportional Fett`

Dies wird verwendet, um Systemeingaben hervorzuheben, einschließlich Shell-Befehle, Dateinamen und -pfade. Es wird ebenfalls zum Hervorheben von Tasten und Tastenkombinationen verwendet. Zum Beispiel:

> Um den Inhalt der Datei **`my_next_bestselling_novel`** in Ihrem aktuellen Arbeitsverzeichnis zu sehen, geben Sie den Befehl **`cat my_next_bestselling_novel`** in den Shell-Prompt ein und drücken Sie **`Enter`**, um den Befehl auszuführen.

Das oben aufgeführte Beispiel beinhaltet einen Dateinamen, einen Shell-Befehl und eine Taste. Alle werden nichtproportional fett dargestellt und alle können, dank des Kontextes, leicht unterschieden werden.

Tastenkombinationen unterscheiden sich von einzelnen Tasten durch das Pluszeichen, das die einzelnen Teile einer Tastenkombination miteinander verbindet. Zum Beispiel:

> Drücken Sie **`Enter`**, um den Befehl auszuführen.
>
> Drücken Sie **`Strg`**+**`Alt`**+**`F2`**, um zu einem virtuellen Terminal zu wechseln.

Das erste Beispiel hebt die zu drückende Taste hervor. Das zweite Beispiel hebt eine Tastenkombination hervor: eine Gruppe von drei Tasten, die gleichzeitig gedrückt werden müssen.

Falls Quellcode diskutiert wird, werden Klassennamen, Methoden, Funktionen, Variablennamen und Rückgabewerte, die innerhalb eines Abschnitts erwähnt werden, wie oben gezeigt **`nichtproportional fett`** dargestellt. Zum Beispiel:

> Zu dateiverwandten Klassen zählen **`filesystem`** für Dateisysteme, **`file`** für Dateien und **`dir`** für Verzeichnisse. Jede Klasse hat ihren eigenen Satz an Berechtigungen.

Proportional Fett

Dies kennzeichnet Wörter oder Sätze, die auf einem System vorkommen, einschließlich Applikationsnamen, Text in Dialogfeldern, beschriftete Schaltflächen, Bezeichnungen für Auswahlkästchen und Radio-Buttons, Überschriften von Menüs und Untermenüs. Zum Beispiel:

> Wählen Sie **System** → **Einstellungen** → **Maus** in der Hauptmenüleiste aus, um die

> **Mauseinstellungen** zu öffnen. Wählen Sie im Reiter **Tasten** auf das Auswahlkästchen **Mit links bediente Maus** und anschließend auf **Schließen**, um die primäre Maustaste von der linken auf die rechte Seite zu ändern (d.h., um die Maus auf Linkshänder anzupassen).
>
> Um ein Sonderzeichen in eine **gedit**-Datei einzufügen, wählen Sie **Anwendungen** → **Zubehör** → **Zeichentabelle** aus der Hauptmenüleiste. Wählen Sie als Nächstes **Suchen** → **Suchen** aus der Menüleiste der **Zeichentabelle**, geben Sie im Feld **Suchbegriff** den Namen des Zeichens ein und klicken Sie auf **Weitersuchen**. Das gesuchte Zeichen wird daraufhin in der **Zeichentabelle** hervorgehoben. Doppelklicken Sie auf dieses hervorgehobene Zeichen, um es in das Feld **Zu kopierender Text** zu übernehmen und klicken Sie anschließend auf die Schaltfläche **Kopieren**. Gehen Sie nun zurück in Ihr Dokument und wählen Sie **Bearbeiten** → **Einfügen** aus der **gedit**-Menüleiste.

Der oben aufgeführte Text enthält Applikationsnamen, systemweite Menünamen und -elemente, applikationsspezifische Menünamen sowie Schaltflächen und Text innerhalb einer grafischen Oberfläche. Alle werden proportional fett dargestellt und sind anhand des Kontextes unterscheidbar.

Nichtproportional Fett Kursiv oder ***Proportional Fett Kursiv***

Sowohl bei nichtproportional fett als auch bei proportional fett weist ein zusätzlicher Kursivdruck auf einen ersetzbaren oder variablen Text hin. Kursivdruck kennzeichnet Text, der nicht wörtlich eingeben wird, oder angezeigten Text, der sich abhängig von den gegebenen Umständen unterscheiden kann. Zum Beispiel:

> Um sich mit einer Remote-Maschine via SSH zu verbinden, geben Sie an einem Shell-Prompt **ssh *username@domain.name*** ein. Falls die Remote-Maschine **example.com** ist und Ihr Benutzername auf dieser Maschine John lautet, geben Sie also **ssh john@example.com** ein.
>
> Der Befehl **mount -o remount *file-system*** hängt das angegebene Dateisystem wieder ein. Um beispielsweise das **/home**-Dateisystem wieder einzuhängen, verwenden Sie den Befehl **mount -o remount /home**.
>
> Um die Version des derzeit installierten Pakets zu sehen, verwenden Sie den Befehl **rpm -q *package***. Die Ausgabe sieht wie folgt aus: ***package-version-release***.

Beachten Sie die kursiv dargestellten Begriffe oben — username, domain.name, file-system, package, version und release. Jedes Wort ist ein Platzhalter entweder für Text, den Sie für einen Befehl eingeben, oder für Text, der vom System angezeigt wird.

Neben der Standardbenutzung für die Darstellung des Titels eines Werks zeigt der Kursivdruck auch die erstmalige Verwendung eines neuen und wichtigen Begriffs an. Zum Beispiel:

> Publican ist ein *DocBook* Publishing-System.

1.2. Konventionen für Seitenansprachen

Ausgaben des Terminals und Auszüge aus dem Quellcode werden visuell vom umliegenden Text hervorgehoben durch sogenannte Seitenansprachen (auch Pull-Quotes genannt).

Eine an das Terminal gesendete Ausgabe wird in den Schrifttyp **nichtproportional Roman** gesetzt und wie folgt dargestellt:

```
books         Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests   Desktop1  downloads      images  notes  scripts  svgs
```

Auszüge aus dem Quellcode werden ebenfalls in den Schrifttyp **`nichtproportional Roman`** gesetzt, doch wird zusätztlich noch die Syntax hervorgehoben:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

1.3. Anmerkungen und Warnungen

Zu guter Letzt verwenden wir drei visuelle Stile, um die Aufmerksamkeit auf Informationen zu lenken, die andernfalls vielleicht übersehen werden könnten.

Anmerkung

Eine Anmerkung ist ein Tipp, ein abgekürztes Verfahren oder ein alternativer Ansatz für die vorliegende Aufgabe. Das Ignorieren von Anmerkungen sollte keine negativen Auswirkungen haben, aber Sie verpassen so vielleicht einen Trick, der Ihnen das Leben vereinfachen könnte.

Wichtig

Die Wichtig-Schaukästen lenken die Aufmerksamkeit auf Dinge, die sonst leicht übersehen werden können: Konfigurationsänderungen, die nur für die aktuelle Sitzung gelten oder Dienste, für die ein Neustart nötig ist, bevor eine Aktualisierung wirksam wird. Das Ignorieren von Wichtig-Schaukästen würde keinen Datenverlust verursachen, kann aber unter Umständen zu Ärgernissen und Frustration führen.

Warnung

Eine Warnung sollte nicht ignoriert werden. Das Ignorieren von Warnungen führt mit hoher Wahrscheinlichkeit zu Datenverlust.

2. Wir freuen uns auf Ihr Feedback!

Falls Sie einen Fehler in diesem Handbuch finden oder eine Idee haben, wie dieses verbessert werden könnte, freuen wir uns über Ihr Feedback! Bitte reichen Sie einen Fehlerbericht in Bugzilla (http://bugzilla.redhat.com/) für das Produkt **Red Hat Enterprise Linux** ein.

Vergewissern Sie sich beim Einreichen eines Fehlerberichts, dass Sie die Kennung des Handbuchs mit angeben: doc-*Security_Guide* sowie die Versionsnummer: **6**.

Falls Sie uns einen Vorschlag zur Verbesserung der Dokumentation senden möchten, sollten Sie hierzu möglichst genaue Angaben machen. Wenn Sie einen Fehler gefunden haben, geben Sie bitte die Nummer des Abschnitts und einen Ausschnitt des Textes an, damit wir diesen leicht finden können.

Kapitel 1. Überblick über Sicherheit

Durch die wachsende Abhängigkeit von leistungsstarken, vernetzten Computern für das Führen von Unternehmen und Aufzeichnen unserer persönlichen Daten haben sich ganze Industriezweige um die Netzwerk- und Computersicherheit herum gebildet. Unternehmen ziehen das Wissen und die Fähigkeiten von Sicherheitsexperten zu Rate, um Systeme zu prüfen und maßgeschneiderte Lösungen für die Anforderungen des Unternehmens zu erstellen. Dadurch, dass die meisten Unternehmen dynamisch arbeiten, mit Mitarbeitern, die auf IT-Ressourcen der Firma intern und extern zugreifen, wird der Bedarf an sicheren EDV-Umgebungen immer deutlicher.

Leider betrachten viele Unternehmen (sowie auch Einzelbenutzer) die Sicherheit immer erst im Nachhinein, ein Faktor, der zu Gunsten erhöhter Leistung und Produktivität sowie aus Kostengründen gerne übersehen wird. Angemessene Sicherheitsimplementierung wird oftmals *postmortem* durchgeführt — erst nachdem ein unberechtigter Zugriff erfolgte. Sicherheitsexperten sind sich einig, dass das Ergreifen richtiger Maßnahmen vor dem Verbinden mit einem nicht vertrauenswürdigen Netzwerk wie dem Internet ein sicheres Mittel zum Verhindern von unerlaubten Zugriffen ist.

Anmerkung

Dieses Handbuch verweist an einigen Stellen auf Dateien im **`/lib`**-Verzeichnis. Wenn Sie 64-bit Systeme verwenden, befinden sich einige der genannten Dateien stattdessen in **`/lib64`**.

1.1. Einführung in Sicherheit

1.1.1. Definition von Computersicherheit

Computersicherheit ist ein allgemeiner Begriff, der einen weitreichenden Bereich der Datenverarbeitung umfasst. Industriezweige, die für ihre täglichen Geschäftstransaktionen und Zugriffe auf wichtige Daten auf Computersysteme und Netzwerke angewiesen sind, betrachten ihre Daten als einen wichtigen Teil ihres Gesamtkapitals. Mehrere Begriffe und Kennzahlen haben ihren Weg in die Geschäftssprache gefunden, wie zum Beispiel Total Cost of Ownership (TCO), Return on Investment (ROI) und Quality of Service (QoS). Anhand dieser Kennzahlen kalkulieren Unternehmen Aspekte wie Datenintegrität und Hochverfügbarkeit als Teil ihrer Planung und Prozessverwaltung. In einigen Industriezweigen wie zum Beispiel dem E-Commerce kann die Verfügbarkeit und Vertrauenswürdigkeit von Daten über Erfolg oder Misserfolg des Unternehmens entscheiden.

1.1.1.1. Anfänge der Computersicherheit

Die Datensicherheit hat sich in den letzten Jahren in Anbetracht der wachsenden Abhängigkeit von öffentlichen Netzwerken für persönliche, finanzielle und andere vertrauliche Informationen entwickelt. Die zahlreichen Fälle, wie z. B. der Mitnick [1] oder der Vladimir Levin [2] Fall, haben Unternehmen aller Industriebereiche dazu veranlasst, ihre Methoden zur Datenübertragung und -aufbewahrung neu zu überdenken. Die wachsende Beliebtheit des Internets war eine der wichtigsten Entwicklungen, die intensivere Bemühungen im Bereich der Datensicherheit mit sich brachte.

Eine stetig wachsende Zahl von Nutzern verwenden ihre eigenen Computer für den Zugriff auf Ressourcen, die das Internet zu bieten hat. Von Recherchen und Informationssuche bis hin zu E-Mail und Handelstransaktionen - das Internet gilt als eine der bedeutendsten Entwicklungen des 20. Jahrhunderts.

Das Internet und seine früheren Protokolle wurden jedoch als ein System auf Vertrauensbasis entwickelt. Mit anderen Worten, das Internetprotokoll (IP) war von vornherein nicht als sicher ausgelegt.

Es sind keine anerkannten Sicherheitsstandards im TCP/IP-Kommunikationsstapel integriert, was eine Angriffsfläche für potenziell böswillige Benutzer und Prozesse im gesamten Netzwerk bildet. Moderne Entwicklungen haben die Kommunikation über das Internet zwar sicherer gemacht, allerdings kommt es immer wieder zu Vorfällen, die Aufsehen erregen und uns bewusst machen, dass nichts hundertprozentig sicher ist.

1.1.1.2. Heutige Sicherheit

Im Februar 2000 wurde ein Distributed Denial of Service (DDoS) Angriff auf einige der am häufigsten besuchten Internetsites ausgeführt. Durch diesen Angriff waren yahoo.com, cnn.com, amazon.com, fbi.gov und einige andere Sites für normale Benutzer unerreichbar, da Router mit stundenlangen, riesigen ICMP-Paketübertragungen, auch *Ping Flood* genannt, überlastet waren. Diese Attacke wurde von unbekannten Angreifern gestartet, die speziell dafür erstellte, einfach erhältliche Programme verwendeten, die angreifbare Netzwerkserver suchen und dann sogenannte *Trojaner*-Client-Applikationen auf den Servern installieren, um schließlich eine zeitlich koordinierte Attacke zu starten, bei der die Site des Opfers durch jeden dieser infizierten Server mit Anfragen überflutet wird und somit unerreichbar wird. Viele sehen die Ursache dieses Angriffs in fundamentalen Fehlern in der Weise, wie Router und Protokolle strukturiert sind, um alle eingehenden Daten anzunehmen, egal woher oder zu welchem Zweck Pakete gesendet wurden.

Im Jahre 2007 führte ein Verstoß gegen die Datensicherheit, der eine bekannte Schwachstelle des Wired Equivalent Privacy (WEP) Protokolls zur Verschlüsselung von Funkverbindungen ausnutzte, zum Diebstahl von über 45 Millionen Kreditkartennummern von einem globalen Finanzinstitut. [3]

In einem anderen Fall wurden die Abrechnungsunterlagen von über 2,2 Millionen Patienten, die auf einem Backup-Band gespeichert waren, vom Beifahrersitz eines Kurierfahrzeugs gestohlen. [4]

Geschätzte 1,4 Milliarden Menschen weltweit nutzen derzeit das Internet oder haben es genutzt. [5] Gleichzeitig wissen wir jedoch auch Folgendes:

- Jeden Tag werden etwa 225 schwerwiegende Fälle von Sicherheitsverletzungen an das CERT-Koordinationszentrum an der Carnegie Mellon Universität gemeldet. [6]
- Die Anzahl der bei CERT gemeldeten Vorfälle stieg sprunghaft von 52.658 im Jahre 2001 auf 82.094 in 2002 und auf 137.529 in 2003 an. [7]
- Laut dem FBI wurde der Schaden durch Computerkriminalität für US-amerikanische Unternehmen für das Jahr 2006 auf 67,2 Milliarden US-Dollar geschätzt. [8]

Eine globale Befragung unter Sicherheits- und Informationstechnologie-Experten im Jahre 2009, "Why Security Matters Now" [9], durchgeführt vom *CIO Magazine*, brachte in diesem Zusammenhang einige bemerkenswerte Ergebnisse zu Tage:

- Nur 23% der Befragten haben Richtlinien zur Verwendung von Web 2.0 Technologien. Diese Technologien wie z. B. Twitter, Facebook und LinkedIn bieten zwar einen bequemen Weg für Unternehmen und Privatpersonen zur Kommunikation und Zusammenarbeit, öffnen gleichzeitig aber auch neue Schwachstellen, insbesondere das mögliche Durchsickern vertraulicher Daten.
- Sogar während der kürzlichen Finanzkrise in 2009 waren die in der Befragung festgestellten Sicherheitsbudgets im Vergleich zu den Vorjahren etwa gleich geblieben oder gestiegen (fast 2 von 3 Befragten erwarteten gleichbleibende oder steigende Ausgaben). Das sind gute Neuigkeiten, da es den Wert widerspiegelt, den Unternehmen heutzutage auf Datensicherheit legen.

Diese Ergebnisse unterstreichen die Tatsache, dass Computersicherheit mittlerweile eine messbare und gerechtfertigte Ausgabe in IT-Budgets ist. Unternehmen, die auf die Datenintegrität und Hochverfügbarkeit angewiesen sind, nehmen die Kenntnisse und Fähigkeiten von

Systemadministratoren, Entwicklern und Technikern in Anspruch, um die Zuverlässigkeit ihrer Systeme, Dienste und Daten rund um die Uhr zu gewährleisten. Böswillige Benutzer, schädliche Prozesse oder koordinierte Angriffe sind eine direkte Bedrohung für den Erfolg eines Unternehmens.

Leider kann die System- und Netzwerksicherheit ein schwieriges Thema sein, welches zudem detailliertes Wissen darüber erfordert, wie ein Unternehmen seine Daten betrachtet, nutzt, bearbeitet und überträgt. Ein Verständnis davon, wie ein Unternehmen (und die Menschen in diesem Unternehmen) seine Geschäfte tätigt ist daher von höchster Bedeutung, um einen angemessenen Sicherheitsplan zu implementieren.

1.1.1.3. Standardisierung der Sicherheit

Unternehmen in jedem Industriezweig sind auf Richtlinien und Regeln von Standardisierungsorganisationen wie z. B. der American Medical Association (AMA) oder dem Institute of Electrical and Electronics Engineers (IEEE) angewiesen. Die gleichen Ideale gelten für die Datensicherheit. Viele Sicherheitsberater und Hersteller haben sich auf das Standard-Sicherheitsmodell CIA (Confidentiality, Integrity und Availability - Vertraulichkeit, Integrität und Verfügbarkeit) geeinigt. Dieses 3-Schichten-Modell ist eine allgemein anerkannte Komponente für das Einschätzen von Risiken für sensible Daten und das Einrichten einer Sicherheitsrichtlinie. Im Folgenden wird das CIA-Modell näher beschrieben:

- Vertraulichkeit — Vertrauliche Informationen dürfen nur für im vornherein festgelegte Einzelpersonen verfügbar sein. Unautorisierte Übertragung und Verwendung von Informationen muss verhindert werden. So stellt zum Beispiel die Vertraulichkeit von Informationen sicher, dass persönliche oder finanzielle Details von Kunden nicht von Unbefugten für böswillige Zwecke wie Identitätsraub oder Kreditbetrug missbraucht werden können.
- Integrität — Informationen dürfen nicht derart verändert werden, dass sie unvollständig oder falsch werden. Unbefugte müssen daran gehindert werden, vertrauliche Informationen ändern oder zerstören zu können.
- Verfügbarkeit — Informationen müssen jederzeit für befugte Personen zugänglich sein. Verfügbarkeit ist die Garantie dafür, dass Informationen mit einer vereinbarten Häufigkeit und rechtzeitig abgerufen werden können. Dies wird häufig in Prozent gemessen und formell in Service-Level-Agreements (SLAs) zwischen Netzwerkservice-Anbietern und deren Geschäftskunden festgelegt.

1.1.2. SELinux

Red Hat Enterprise Linux beinhaltet eine Erweiterung zum Linux-Kernel namens SELinux, die eine Mandatory Access Control (MAC) Architektur implementiert, welche feingranulare Kontrolle über Dateien, Prozesse, Benutzer und Applikationen im System ermöglicht. Eine detaillierte Auseinandersetzung mit SELinux geht über den Rahmen dieses Handbuchs hinaus. Werfen Sie für mehr Informationen über SELinux und dessen Anwendung in Red Hat Enterprise Linux bitte einen Blick auf das Red Hat Enterprise Linux SELinux-Benutzerhandbuch. Weitere Informationen über das Konfigurieren und Ausführen von Diensten, die durch SELinux gesichert sind, finden Sie im SELinux-Handbuch zur Verwaltung eingeschränkter Dienste. Andere verfügbare Quellen für SELinux finden Sie unter Kapitel 8, *Weitere Informationsquellen*.

1.1.3. Sicherheitskontrollen

Computersicherheit wird häufig in drei verschiedene Hauptkategorien eingeteilt, die allgemein als *Kontrollen* bezeichnet werden:

- Physische Kontrolle
- Technische Kontrolle
- Administrative Kontrolle

Diese drei Kategorien definieren die Hauptziele einer ordnungsgemäßen Sicherheitsimplementierung. Innerhalb dieser Kontrollen befinden sich Unterkategorien, die deren Implementierung näher definieren.

1.1.3.1. Physische Kontrolle

Die physische Kontrolle ist die Implementierung von Sicherheitsmaßnahmen in einer festgelegten Struktur, die unbefugten Zugriff auf sensible Daten verhindert. Beispiele für physische Zugangskontrollen:

- Überwachungskameras
- Bewegungs- oder Wärmemelder
- Sicherheitspersonal
- Ausweise
- Verriegelte Stahltüren
- Biometrie (z. B. Erkennung von Fingerabdrücken, Stimme, Gesicht, Iris, Handschrift oder andere automatisierte Methoden, um die Identität von Personen nachzuweisen)

1.1.3.2. Technische Kontrollen

Technische Kontrollen verwenden Technologie als Basis für die Kontrolle von Zugang zu bzw. Verwendung von sensiblen Daten durch eine physische Struktur und über ein Netzwerk. Technische Kontrollen decken weite Bereiche ab und umfassen unter anderem folgende Technologien:

- Verschlüsselung
- Smart Cards
- Netzwerkauthentifizierung
- Zugangskontrolllisten (ACLs)
- Software zur Prüfung der Dateiintegrität

1.1.3.3. Administrative Kontrollen

Administrative Kontrollen definieren den menschlichen Faktor der Sicherheit. Sie umfassen alle Mitarbeiter innerhalb eines Unternehmens und legen fest, welche Benutzer Zugang zu welchen Ressourcen und Informationen haben. Dies geschieht unter anderem durch:

- Schulung und Aufklärung
- Katastrophenvorbereitung und Wiederherstellungspläne
- Personaleinstellungs- und Separations-Strategien
- Mitarbeiterregistrierung und Buchhaltung

1.1.4. Fazit

Nachdem Sie jetzt mehr über die Ursprünge, Beweggründe und Aspekte der Sicherheit erfahren haben, können Sie nun den richtigen Aktionsplan für Red Hat Enterprise Linux festlegen. Es ist wichtig zu wissen, welche Faktoren und Bedingungen die Sicherheit beinflussen, um eine richtige Strategie planen und implementieren zu können. Mit diesen Informationen im Hinterkopf kann der Prozess formalisiert werden, und der Weg wird klarer, je tiefer Sie in die Details des Sicherheitsprozesses eintauchen.

1.2. Schwachstellenanalyse

Mit genügend Zeit, Ressourcen und Motivation kann ein Angreifer in fast jedes System einbrechen. Schlussendlich bieten alle derzeit erhältlichen Technologien und Sicherheitsprozeduren keine Garantie dafür, dass ein System vor Eindringlingen vollkommen sicher ist. Router können bei der Sicherung Ihrer Gateways zum Internet helfen. Firewalls helfen bei der Sicherung des Netzwerks. Virtuelle Private

Netzwerke können auf sichere Art Daten verschlüsselt übertragen. Intrusion-Detection-Systeme können Sie vor böswilligen Aktivitäten warnen. Der Erfolg jeder dieser Technologien hängt jedoch von einer Reihe von Variablen ab. Diese sind unter anderem:

- Die Kompetenz der Mitarbeiter, die für die Konfiguration, Überwachung und Wartung dieser Technologien verantwortlich sind.
- Die Fähigkeit, Dienste und Kernel schnell und effizient mit Patches versehen und aktualisieren zu können.
- Die Fähigkeit der Verantwortlichen, konstante Wachsamkeit im Netzwerk auszuüben.

Durch die Dynamik von Datensystemen und Technologien kann das Sichern Ihrer Ressourcen ziemlich komplex werden. Aufgrund dieser Komplexität kann es sich schwierig gestalten, Experten für Ihre Systeme zu finden. Es ist zwar möglich, Mitarbeiter mit reichhaltigem Wissen auf vielen Gebieten der Informationssicherheit zu beschäftigen, aber es ist relativ schwierig, Experten auf mehr als nur wenigen Gebieten zu finden. Dies liegt hauptsächlich daran, dass die Informationssicherheit ständige Aufmerksamkeit verlangt. Die Informationssicherheit befindet sich in stetigem Wandel.

1.2.1. Denken wie der Feind

Angenommen, Sie verwalten ein Firmennetzwerk. Solche Netzwerke bestehen meistens aus Betriebssystemen, Applikationen, Servern, Netzwerküberwachung, Firewalls, Intrusion-Detection-Systemen und vielem mehr. Stellen Sie sich jetzt vor, Sie müssen dahingehend immer auf dem neuesten Stand sein. Durch die Komplexität heutiger Software und Netzwerkumgebungen sind Angriffe auf einen Rechner unter Ausnutzung einer Sicherheitslücke oder eines Bugs beinahe gewiss. Mit allen Patches und Updates für ein gesamtes Netzwerk auf dem Laufenden zu sein, ist eine gewaltige Aufgabe innerhalb eines großen Unternehmens mit heterogenen Systemen.

Wenn Sie nun diese gewaltigen Anforderungen an das Wissen mit der Aufgabe, immer auf dem neuesten Stand zu sein, kombinieren, sind Systemeinbrüche, Datenkorruption, Serviceunterbrechungen und andere Vorfälle unvermeidbar.

Um den Nutzen von Sicherheitstechnologien zu erhöhen und dabei zu helfen, Systeme, Netzwerke und Daten zu schützen, sollten Sie sich in die Lage eines Angreifers versetzen und die Sicherheit der Systeme durch das Suchen von Schwachstellen testen. Vorbeugende Schwachstellenanalysen für Ihre eigenen Systeme und Netzwerkressourcen können potenzielle Problemstellen aufdecken, bevor ein Angreifer diese zu seinem Vorteil ausnutzen kann.

Eine Schwachstellenanalyse ist eine interne Prüfung Ihrer Netzwerk- und Systemsicherheit. Die Ergebnisse zeigen die Vertraulichkeit, Integrität und Verfügbarkeit Ihres Netzwerks auf (wie in Abschnitt 1.1.1.3, „Standardisierung der Sicherheit“ beschrieben). Eine Schwachstellenanalyse beginnt für gewöhnlich mit einer Erkundungsphase, in der wichtige Daten zum System und Ressourcen gesammelt werden. Diese Phase führt zur Systembereitschaftsphase, in der das Zielsystem auf alle bekannten Schwachstellen hin geprüft wird. Diese Phase führt dann zur Berichterstattungsphase, in der die Ergebnisse in die Risikokategorien Hoch, Mittel und Niedrig eingestuft und Methoden zur Verbesserung der Sicherheit (oder Schwächung der Anfälligkeit) des Zielsystems diskutiert werden.

Würden Sie zum Beispiel eine Schwachstellenanalyse für Ihr Haus durchführen, würden Sie wahrscheinlich prüfen, ob jede Tür geschlossen und verriegelt ist. Sie würden auch jedes Fenster prüfen und sicherstellen, dass diese richtig geschlossen und verriegelt sind. Das gleiche Prinzip gilt auch für Systeme, Netzwerke und elektronische Daten. Benutzer mit böswilligen Absichten sind die Diebe und Vandalen Ihrer Daten. Konzentrieren Sie sich auf deren Tools, Mentalität und Beweggründe, denn so können Sie schnell auf deren Taten reagieren.

1.2.2. Definition von Analyse und Test

Schwachstellenanalysen können in zwei Arten klassifiziert werden: *von außen hineinspähen* und *innen herumschnüffeln*.

Wenn Sie eine Schwachstellenanalyse von außen betrachtet durchführen, so versuchen Sie, Ihr System von außen zu kompromittieren. Wenn Sie Ihr Unternehmen von extern betrachten, versetzen Sie sich in die Sichtweise eines Crackers. Sie sehen, was der Cracker sehen kann — öffentlich-weiterleitbare IP-Adressen, Systeme in Ihrer *DMZ*, externe Schnittstellen Ihrer Firewall und vieles mehr. DMZ steht für "Demilitarized Zone", was einen Computer oder ein kleines Subnetzwerk bezeichnet, das sich zwischen einem internen, zuverlässigen Netzwerk, wie z. B. einem gemeinschaftlichen privaten LAN und einem unzuverlässigen, externen Netzwerk, wie z. B. dem öffentlichen Internet, befindet. Üblicherweise beinhaltet die DMZ Geräte, die für den Internetverkehr zugänglich sind, wie z. B. Web (HTTP)-Server, FTP-Server, SMTP (E-Mail)-Server und DNS-Server.

Wenn Sie eine Schwachstellenanalyse von innen betrachtet durchführen, haben Sie den gewissen Vorteil, das Sie bereits intern sind, und Sie einen Status als "vertrauenswürdig" haben. Dies ist der Blickwinkel, den Sie und Ihre Kollegen haben, wenn Sie sich einmal im System angemeldet haben. Sie sehen Druckserver, Dateiserver, Datenbanken und andere Ressourcen.

Es gibt klare Unterschiede zwischen diesen beiden Arten der Schwachstellenanalyse. Als interner Mitarbeiter Ihres Unternehmens besitzen Sie höhere Privilegien, weit mehr als jeder Außenstehende. Entsprechend sind die Sicherheitsrichtlinien in den meisten Unternehmen nach wie vor so konfiguriert, externe Eindringlinge fernzuhalten. Es wird nur sehr wenig für die interne Sicherung des Unternehmens getan (z. B. Firewalls für Abteilungen, Zugangskontrollen auf Benutzerebene, Authentifizierungsvorgänge für interne Ressourcen und so weiter). Üblicherweise gibt es wesentlich mehr Ressourcen, wenn man sich intern umschaut, da die meisten Systeme in einem Unternehmen intern sind. Sobald Sie sich einmal außerhalb eines Unternehmens befinden, erhalten Sie sofort den Status "nicht vertrauenswürdig". Die extern zugänglichen Systeme und Ressourcen sind für gewöhnlich wesentlich stärker eingeschränkt.

Beachten Sie die Unterschiede zwischen Schwachstellenanalyse und *Penetration Test*. Sehen Sie die Schwachstellenanalyse als ersten Schritt zu einem Penetration Test an. Die Informationen aus der Schwachstellenanalyse werden im Test angewendet. Mit der Analyse wird nach Lücken und möglichen Schwachstellen im System gesucht, während der Penetration Test die Ergebnisse in die Tat umsetzt.

Die Einschätzung der Netzwerkinfrastruktur ist ein dynamischer Prozess. Sowohl Informationssicherheit als auch physische Sicherheit ist dynamisch. Das Durchführen der Analyse gibt einen Überblick, kann jedoch auch falsche Ergebnisse liefern.

Sicherheitsadministratoren sind nur so gut wie die Tools, die diese benutzen, und das Wissen, das diese besitzen. Nehmen Sie eines der aktuell erhältlichen Analyse-Tools und lassen Sie es über Ihr System laufen. Dabei ist fast garantiert, dass einige Schwachstellen gefunden werden, die gar nicht existieren. Ob durch einen Programmfehler oder Benutzerfehler hervorgerufen, das Ergebnis ist das gleiche: Das Tool findet Schwachstellen, die gar nicht existieren, oder schlimmer noch, es findet wirklich existierende Schwachstellen nicht.

Da wir nun den Unterschied zwischen Schwachstellenanalyse und Penetration Test definiert haben, ist es ratsam, die Ergebnisse der Analyse sorgfältig zu prüfen, bevor Sie den Penetration Test tatsächlich durchführen.

Warnung

Der Versuch, Schwachstellen in Produktionsressourcen aufzudecken, kann einen negativen Effekt auf die Produktivität und Effizienz Ihrer Systeme und Netzwerke haben.

In der folgenden Liste werden einige der Vorteile einer Schwachstellenanalyse aufgezeigt

- Proaktiver Fokus auf Informationssicherheit
- Auffinden potenzieller Schwachstellen, bevor diese von Angreifern gefunden werden
- Resultiert normalerweise darin, dass Systeme aktuell gehalten und mit Patches versehen werden
- Fördert Wachstum und hilft bei der Entwicklung von Mitarbeiterkompetenz
- Vermindert finanzielle Verluste und negative Presse

1.2.2.1. Entwickeln einer Methodik

Um die Auswahl der richtigen Tools für die Schwachstellenanalyse zu unterstützen, ist es sinnvoll, zuerst eine Methodik für die Schwachstellenanalyse zu entwickeln. Es gibt zur Zeit leider keine vordefinierte oder industrieweit bewährte Methodik, jedoch reichen meistens gesunder Menschenverstand und empfohlene Verfahren als Leitfaden aus.

Was ist das Ziel? Betrachten wir nur einen Server, oder das gesamte Netzwerk und alles innerhalb des Netzwerks? Betrachten wir das Unternehmen intern oder extern? Die Antworten auf diese Fragen sind wichtig, da diese Ihnen bei der Entscheidung über die richtigen Tools und deren Einsatz helfen.

Weitere Informationen zur Entwicklung von Methodiken finden Sie auf den folgenden Websites:

- http://www.isecom.org/osstmm/ *The Open Source Security Testing Methodology Manual* (OSSTMM)
- http://www.owasp.org/ *The Open Web Application Security Project*

1.2.3. Bewerten der Tools

Eine typische Analyse beginnt mit dem Einsatz eines Tools für das Sammeln von Informationen. Bei der Analyse des gesamten Netzwerks sollten Sie zuerst das Layout festlegen, um aktive Hosts zu identifizieren. Sobald diese gefunden wurden, sollten Sie jeden Host einzeln untersuchen. Das Untersuchen dieser Hosts bedarf weiterer Tools. Das Wissen, welche Tools für was verwendet werden, ist der wohl bedeutendste Schritt beim Aufdecken von Schwachstellen.

Wie in jedem Bereich des täglichen Lebens gibt es viele verschiedene Tools, die die gleiche Arbeit verrichten. Dies trifft auch auf Schwachstellenanalysen zu. Es gibt Tools, die speziell für Betriebssysteme, Applikationen oder Netzwerke (basierend auf den verwendeten Protokollen) eingesetzt werden können. Einige Tools sind kostenlos, andere wiederum nicht. Einige Tools sind intuitiv und benutzerfreundlich, andere eher kryptisch und schlecht dokumentiert aber besitzen Features, die andere Tools wiederum nicht haben.

Die Suche der richtigen Tools kann eine Herausforderung sein; schlussendlich zählt die Erfahrung. Wenn möglich, richten Sie ein Testlabor ein und probieren so viele Tools aus wie nur möglich, und beachten Sie dabei die Stärken und Schwächen. Lesen Sie die README-Datei oder Handbuchseite zum Tool. Suchen Sie zusätzlich dazu im Internet nach weiteren Informationen wie Artikel, Schritt-für-Schritt-Anleitungen und Mailing-Listen für ein Tool.

Die unten beschriebenen Tools sind nur wenige Beispiele für die erhältlichen Tools.

1.2.3.1. Scannen von Hosts mit Nmap

Nmap ist ein beliebtes Tool, das zum Feststellen eines Netzwerk-Layouts verwendet werden kann. Nmap ist schon seit vielen Jahren auf dem Markt und ist das wahrscheinlich am häufigsten verwendete Tool zum Sammeln von Informationen. Es enthält eine ausgezeichnete Handbuchseite, die detaillierte Informationen zu Optionen und zur Verwendung bietet. Administratoren können Nmap in einem Netzwerk verwenden, um Hosts und offene Ports auf diesen Systemen zu finden.

Nmap ist ein kompetenter, erster Schritt bei der Schwachstellenanalyse. Sie können die Hosts in Ihrem

Netzwerk aufzeigen und eine Option angeben, die versucht zu bestimmen, welches Betriebssystem auf einem bestimmten Host läuft. Nmap ist eine gute Grundlage zum Einführen einer Richtlinie, nach der sichere Dienste verwendet werden und unbenutzter Dienste eingeschränkt werden.

1.2.3.1.1. Verwendung von Nmap

Nmap kann von einem Shell-Prompt aus verwendet werden. Geben Sie an einem Shell-Prompt den Befehl **`nmap`** gefolgt vom Hostnamen oder der IP-Adresse des zu scannenden Computers ein.

```
nmap foo.example.com
```

Die Ergebnisse des Scannens (was einige Minuten dauern kann, abhängig davon, wo sich der Host befindet), sollten wie folgt aussehen:

```
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
80/tcp  open   http
113/tcp closed auth
```

Nmap prüft die häufigsten Ports für die Netzwerkkommunikation auf horchende oder wartende Dienste. Dieses Wissen ist sinnvoll für Administratoren, die unnötige Dienste abschalten möchten.

Weitere Informationen zu Nmap finden Sie auf der offiziellen Homepage unter folgender URL:

http://www.insecure.org/

1.2.3.2. Nessus

Nessus ist ein umfassender Sicherheitsscanner. Die Plug-In-Architektur von Nessus ermöglicht Benutzern das Anpassen an deren Systeme und Netzwerke. Wie jeder Scanner ist auch Nessus nur so gut wie die Signatur-Datenbank, die verwendet wird. Glücklicherweise wird Nessus häufig aktualisiert und bietet vollständige Berichterstattung, Host-Scanning und Schwachstellensuche in Echtzeit. Denken Sie jedoch immer daran, dass fehlerhafte Ergebnisse auch bei einem so leistungsstarken und häufig aktualisierten Tool wie Nessus auftreten können.

Anmerkung

Die Software für den Nessus-Client und den Nessus-Server erfordert eine entsprechende Subskription. Die Erwähnung in diesem Handbuch ist nur ein Hinweis für Benutzer, die eventuell an dieser beliebten Applikation interessiert sind.

Weitere Informationen zu Nessus finden Sie auf der offiziellen Homepage unter folgender URL:

http://www.nessus.org/

1.2.3.3. Nikto

Nikto ist ein ausgezeichneter CGI-Scanner (Common Gateway Interface). Nikto hat die Fähigkeit, nicht nur CGI-Schwachstellen zu suchen, sondern diese auch so zu prüfen, dass Intrusion-Detection-Systeme umgangen werden. Es wird von ausgezeichneter Dokumentation begleitet, die vor dem Ausführen des Programms sorgfältig gelesen werden sollte. Wenn Sie Webserver mit CGI-Skripten besitzen, ist Nikto ein ausgezeichneter Scanner zum Prüfen der Sicherheit dieser Server.

Weitere Informationen zu Nikto finden Sie unter folgender URL:

http://cirt.net/nikto2

1.2.3.4. Für Ihre zukünftigen Bedürfnisse vorausplanen

Abhängig von Ihrem Ziel und den Ressourcen gibt es viele Tools auf dem Markt. Es gibt Tools für Wireless-Netzwerke, Novell-Netzwerke, Windows-Systeme, Linux-Systeme und vieles mehr. Ein weiterer, wichtiger Teil der Analysen können auch die physische Sicherheit, Mitarbeiterüberwachung oder Voice/PBX-Netzwerkanalysen sein. Sie können neue Konzepte wie das *War Walking* und *War Driving* — das Scannen der Umgebung der physischen Unternehmensstruktur auf Schwachstellen im Wireless-Netzwerk — erforschen und in Ihre Analysen einbinden. Fantasie und Erfahrung im Umgang mit dem Auffinden und Lösen von Sicherheitsproblemen sind die einzigen Grenzen bei der Planung und Durchführung von Schwachstellenanalysen.

1.3. Angreifer und Schwachstellen

Um eine gute Sicherheitsstrategie planen und implementieren zu können, müssen Sie als Erstes die Schwachstellen verstehen, die entschlossene, motivierte Angreifer ausnutzen könnten, um Systeme zu schädigen. Bevor wir jedoch ins Detail gehen, lassen Sie uns zunächst die Terminologie definieren, die bei der Identifikation eines Angreifers verwendet wird.

1.3.1. Ein kurzer geschichtlicher Überblick über Hacker

Die moderne Bedeutung des Begriffs *Hacker* geht auf die 60er Jahre und den Massachusetts Institute of Technology (MIT) Tech Model Railroad Club zurück, der detailgetreue Modelleisenbahnen in großem Umfang entwickelte. Als Hacker wurden Clubmitglieder bezeichnet, die einen Trick oder eine Lösung für ein Problem gefunden hatten.

Der Begriff Hacker wurde seitdem verwendet, um angefangen von Computerfreaks bis hin zu talentierten Programmierern alles zu beschreiben. Für viele Hacker charakteristisch ist die Bereitschaft, mit nur wenig oder ganz ohne Fremdmotivation im Detail herauszufinden, wie Computersysteme und Netzwerke funktionieren. Open Source Softwareentwickler betrachten sich selbst und ihre Kollegen oftmals als Hacker und verwenden das Wort als Ausdruck von Respekt.

Normalerweise folgen Hacker einer Form von *Hacker-Ethik*, die vorgibt, dass die Suche nach Informationen und Wissen essentiell ist, und dass die Weitergabe dieses Wissens eine Pflicht des Hackers gegenüber der Community ist. Während dieser Suche nach Wissen genießen einige Hacker die intellektuelle Herausforderung, Sicherheitskontrollen für Computersysteme zu umgehen. Aus diesem Grund verwenden die Medien häufig den Begriff Hacker für jemanden, der unberechtigt mit skrupellosen, böswilligen oder kriminellen Absichten auf Systeme und Netzwerke zugreift. Ein zutreffenderer Begriff für diese Art von Computerhacker ist *Cracker* — ein Begriff, der Mitte der 80er Jahre von Hackern geschaffen wurde, um diese beiden Gruppen zu unterscheiden.

1.3.1.1. Grauzonen

Es gibt einige wesentliche Unterschiede zwischen den einzelnen Personengruppen, die Schwachstellen in Systemen und Netzwerken finden und ausnutzen. Diese unterschiedlichen Gruppen werden oft durch die Farbe des Hutes beschrieben, den sie "tragen", während sie ihre Sicherheitsrecherchen durchführen. Die jeweilige Farbe steht für die Absichten dieser Gruppe.

Ein *White Hat Hacker* ist jemand, der Netzwerke und Systeme testet, um deren Leistung zu untersuchen und Anfälligkeiten auf Angriffe herauszufinden. Gewöhnlich greifen White Hat Hackers ihre eigenen Systeme oder die Systeme von Kunden an, von denen sie zum Zwecke der Sicherheitsprüfung beauftragt wurden. Akademische Forscher und professionelle Sicherheitsberater sind zwei Beispiele für

White Hat Hackers.

Ein *Black Hat Hacker* ist synonym mit einem Cracker. Im Allgemeinen konzentrieren sich Cracker weniger auf das Programmieren und die akademische Seite des Einbruchs in Systeme. Sie verlassen sich häufig auf verfügbare Cracking-Programme und nutzen bekannte Schwachstellen in Systemen zur Aufdeckung sensibler Informationen aus, entweder um persönlichen Gewinn daraus zu erzielen oder um Schaden auf dem System oder Netzwerk anzurichten.

Ein *Gray Hat Hacker* dagegen hat in den meisten Fällen die Fähigkeiten und die Absichten eines White Hat Hackers, setzt sein Wissen gelegentlich jedoch auch mit weniger edlen Absichten ein. Ein Gray Hat Hacker kann also als jemand bezeichnet werden, der grundsätzlich die guten Absichten eines White Hat Hackers hat, jedoch manchmal aus Eigennutz zum Black Hat Hacker wird.

Gray Hat Hacker halten sich häufig an eine andere Form von Hacker-Ethik, nach der es akzeptabel ist, in Systeme einzubrechen, solange der Hacker keinen Diebstahl begeht oder den Datenschutz verletzt. Man kann sich jedoch darüber streiten, ob das eigentliche Einbrechen in Systeme nicht bereits unethisch ist.

Unabhängig von der Absicht des Eindringlings ist es wichtig, die Schwachstellen zu kennen, die ein Cracker am ehesten versucht auszunutzen. Das restliche Kapitel behandelt diese Thematik.

1.3.2. Bedrohungen der Netzwerksicherheit

Unzureichende Methoden bei der Konfiguration einiger Netzwerkaspekte kann das Risiko eines Angriffs erheblich erhöhen.

1.3.2.1. Unsichere Architekturen

Ein fehlerhaft konfiguriertes Netzwerk ist ein Hauptangriffspunkt für unbefugte Benutzer. Ein offenes, lokales Netzwerk ungeschützt dem höchst unsicheren Internet auszusetzen, ist vergleichbar damit, Ihre Haustür in einem unsicheren Stadtteil offen zu lassen — für eine Weile mag nichts passieren, aber *irgendwann* wird sich jemand die Gelegenheit zu Nutze machen.

1.3.2.1.1. Broadcast-Netzwerke

Systemadministratoren unterschätzen oftmals die Bedeutung der Netzwerk-Hardware in ihren Sicherheitssystemen. Einfache Hardware wie z. B. Hubs und Router arbeiten nach dem Broadcast oder ungeschaltetem Prinzip; d. h. wenn ein Knoten Daten über ein Netzwerk überträgt, sendet der Hub oder der Router die Datenpakete solange, bis der Empfängerknoten die Daten empfangen und verarbeitet hat. Diese Methode ist am anfälligsten für *ARP* (Address Resolution Protocol) oder *MAC* (Media Access Control) Adress-Spoofing sowohl durch Angreifer von außen als auch durch unbefugte Benutzer auf lokalen Hosts.

1.3.2.1.2. Zentralisierte Server

Ein weiterer Fallstrick in Netzwerken ist die Verwendung zentralisierter Rechner. Eine beliebte Maßnahme zur Kostensenkung für Unternehmen ist es, alle Dienste auf einer einzigen, leistungsstarken Maschine zusammenzuführen. Dies ist bequem, da einfacher zu verwalten, und es kostet wesentlich weniger als eine Konfiguration mit mehreren Servern. Ein zentralisierter Server stellt jedoch einen einzelnen Ausfallpunkt im Netzwerk dar. Wird der zentrale Server beschädigt, kann dadurch das gesamte Netzwerk nutzlos oder gar zur Angriffsfläche für Datenmanipulation oder Diebstahl werden. In diesen Fällen wird ein zentraler Server zum offenen Einfallstor und erlaubt Zugang zum gesamten Netzwerk.

1.3.3. Bedrohungen der Serversicherheit

Serversicherheit ist genauso wichtig wie Netzwerksicherheit, da Server meistens einen Großteil der

unternehmenskritischen Informationen enthalten. Wird ein Server angegriffen, kann der Cracker auf den gesamten Inhalt zugreifen und nach Belieben Daten stehlen oder manipulieren. Die folgenden Abschnitte behandeln die wichtigsten Aspekte der Serversicherheit.

1.3.3.1. Unbenutzte Dienste und offene Ports

Eine vollständige Installation von Red Hat Enterprise Linux enthält über 1000 Applikationen und Bibliotheken. Die meisten Systemadministratoren wählen jedoch nicht alle Paket der Distribution zur Installation aus, sondern bevorzugen eine Basisinstallation von Paketen inklusive mehrerer Serverapplikationen.

Systemadministratoren tendieren häufig dazu, das Betriebssystem zu installieren, ohne darauf zu achten, welche Programme eigentlich installiert werden. Dies kann problematisch werden, da eventuell nicht benötigte Dienste installiert werden, die mit den Standardeinstellungen konfiguriert und standardmäßig aktiviert werden. Folglich laufen eventuell unerwünschte Dienste wie Telnet, DHCP oder DNS auf einem Server oder einem Arbeitsplatzrechner, ohne dass der Systemadministrator es merkt, was wiederum zu unerwünschtem Netzwerkverkehr zum Server oder sogar zu einem möglichen Einstiegspunkt für Angreifer führen kann. Weitere Informationen zum Schließen von Ports und Deaktivieren unbenutzter Dienste finden Sie unter Abschnitt 2.2, „Server-Sicherheit“.

1.3.3.2. Dienste ohne Patches

Die meisten Serverapplikationen, die in einer Standardinstallation enthalten sind, sind solide, gründlich getestete Softwareapplikationen. Dadurch, dass diese viele Jahre in Produktionsumgebungen eingesetzt wurden, ist ihr Code ausgereift und viele der Fehler sind gefunden und behoben worden.

So etwas wie perfekte Software gibt es jedoch nicht, es ist immer Raum für weitere Verbesserungen. Des Weiteren ist neuere Software nicht immer so umfassend getestet, wie man erwarten würde, z. B. weil diese erst seit Kurzem in der Produktionsumgebung eingesetzt wird oder weil sie noch nicht so beliebt ist wie andere Serversoftware.

Entwickler und Systemadministratoren finden häufig Schwachstellen in Serverapplikationen und veröffentlichen diese Informationen auf Bug-Tracking-Websites und anderen sicherheitsbezogenen Websites wie die Bugtraq-Mailingliste (http://www.securityfocus.com) oder die Website des Computer Emergency Response Team (CERT) (http://www.cert.org). Auch wenn diese Mechanismen eine effektive Methode zur Warnung der Community vor Sicherheitsproblemen darstellt, liegt es letztendlich an den Systemadministratoren, ihre Systeme sofort mit einem Patch zu versehen. Dies ist insbesondere wichtig, da auch Cracker Zugang zu denselben Tracking-Diensten haben und diese Informationen ausnutzen, um nicht gepatchte Systeme anzugreifen. Eine gute Systemadministration verlangt Wachsamkeit, andauerndes Bug Tracking und vernünftige Systemwartung für eine sichere Rechenumgebung.

Weitere Informationen dazu, wie Sie ein System immer auf dem aktuellsten Stand halten können, finden Sie unter Abschnitt 1.5, „Sicherheitsaktualisierungen“.

1.3.3.3. Unaufmerksame Administration

Administratoren, die ihre Systeme nicht mit den neuesten Patches versehen, stellen eine der größten Bedrohungen für die Serversicherheit dar. Nach Angaben des *SysAdmin, Audit, Network, Security Institute* (*SANS*) liegt der Hauptgrund für Computersicherheitsprobleme darin, "unqualifizierte Mitarbeiter mit der Wartung der Sicherheit zu betrauen, ohne ihnen richtiges Training oder die nötige Zeit zur Verfügung zu stellen, um den Job ordnungsgemäß auszuführen." [10] Dies trifft sowohl auf unerfahrene Administratoren als auch auf vermessene oder unmotivierte Administratoren zu.

Einige Administratoren versäumen es, ihre Server oder Workstations zu patchen, während andere vergessen, Protokollmeldungen vom Systemkernel und den Netzwerkverkehr zu beobachten. Ein weiterer häufiger Fehler besteht darin, die Standardpasswörter oder -schlüssel für Dienste nicht zu

verändern. So haben zum Beispiel einige Datenbanken standardmäßige Administrationspasswörter, weil die Datenbankentwickler annehmen, dass der Systemadministrator diese sofort nach der Installation ändert. Vergisst jedoch ein Systemadministrator, diese Passwörter zu ändern, können sogar unerfahrene Cracker mit einem weitverbreiteten Standardpasswort auf administrative Privilegien dieser Datenbank zugreifen. Dies sind nur einige Beispiele dafür, wie unaufmerksame Administration zu unsicheren Servern führen kann.

1.3.3.4. Von Natur aus unsichere Dienste

Auch das wachsamste Unternehmen kann Opfer von Schwachstellen werden, wenn die gewählten Netzwerkdienste von Natur aus unsicher sind. Es werden zum Beispiel viele Dienste unter der Annahme entwickelt, dass diese über sichere Netzwerke verwendet werden; diese Annahme ist jedoch hinfällig, sobald diese Dienste über das Internet verfügbar gemacht werden — welches selbst von Natur aus unsicher ist.

Eine Art von unsicheren Netzwerkdienste sind solche, die Benutzernamen und Passwörter für die Authentifizierung benötigen, diese Informationen bei der Übertragung über das Netzwerk jedoch nicht verschlüsseln. Telnet und FTP sind solche Dienste. Paket-Sniffing-Software, die den Verkehr zwischen entfernten Benutzern und einem solchen Server überwacht, kann so problemlos die Benutzernamen und Passwörter abfangen.

Die oben genannten Dienste können somit auch leichter einem sogenannten *Man-in-the-Middle*-Angriff zum Opfer fallen. Bei dieser Art von Angriff leitet ein Cracker den Netzwerkverkehr um, indem er einen kompromittierten Name-Server dazu bringt, auf seinen Rechner zu verweisen anstatt auf den richtigen Server. Sobald daraufhin jemand eine Remote-Session zu dem Server öffnet, verhält sich der Rechner des Angreifers als unsichtbare Zwischenleitung, und sitzt dabei unerkannt zwischen dem entfernten Dienst und dem ahnungslosen Benutzer und sammelt Informationen. Auf diese Weise kann ein Angreifer Administrationspasswörter und Daten sammeln, ohne dass der Server oder der Benutzer dies merkt.

Eine weitere Art von unsicheren Diensten sind Netzwerkdateisysteme und Informationssysteme wie zum Beispiel NFS oder NIS, die ausdrücklich für eine Verwendung in LANs entwickelt wurden, dann jedoch unglücklicherweise für WANs erweitert wurden (für entfernte Benutzer). NFS hat standardmäßig keine Authentifizierungs- oder Sicherheitsmechanismen konfiguriert, um Angreifer daran zu hindern, die NFS-Freigabe einzuhängen und Zugang zu sämtlichen Inhalten zu erlangen. NIS verfügt auch über wichtige Informationen, die jedem Computer im Netzwerk bekannt sein müssen, einschließlich Passwörter und Dateiberechtigungen innerhalb einer Nur-Text ASCII oder DBM (ASCII-abgeleiteten) Datenbank. Ein Angreifer, der Zugang zu dieser Datenbank erhält, kann dann auf jedes Benutzerkonto in diesem Netzwerk zugreifen, einschließlich dem des Administrators.

Standardmäßig sind bei Red Hat Enterprise Linux solche Dienste deaktiviert. Da Administratoren häufig jedoch zur Verwendung dieser Dienste gezwungen sind, ist eine sorgfältige Konfiguration entscheidend. Weitere Informationen zum sicheren Einrichten eines Servers finden Sie unter Abschnitt 2.2, „Server-Sicherheit“.

1.3.4. Bedrohungen der Arbeitsplatzrechner- und Heim-PC-Sicherheit

Arbeitsplatzrechner und Heim-PCs sind nicht ganz so anfällig für Angriffe wie Netzwerke oder Server, da sie jedoch häufig sensible Informationen wie zum Beispiel Kreditkartendaten enthalten, werden sie schnell zum Ziel von Crackern. Arbeitsplatzrechner können kooptiert werden, ohne dass der Benutzer dies merkt, und können von Angreifern als "Slave"-Maschinen für koordinierte Angriffe verwendet werden. Aus diesem Grund ist es wichtig, sich der Schwachstellen eines Arbeitsplatzrechners bewusst zu sein, um sich eine nervenaufreibende Neuinstallation eines Betriebssystems oder, schlimmer noch, die Schadensbegrenzung nach einem Datendiebstahl zu ersparen.

1.3.4.1. Unsichere Passwörter

Unsichere Passwörter sind eine der leichtesten Methoden für einen Angreifer, Zugang zu einem System zu erhalten. Weitere Informationen darüber, wie Sie häufige Fehler bei der Passwortwahl vermeiden, finden Sie unter Abschnitt 2.1.3, „Passwortsicherheit“.

1.3.4.2. Anfällige Client-Applikationen

Auch wenn ein Administrator über einen sicheren und gepatchten Server verfügt, heißt dies noch lange nicht, dass Remote-Benutzer sicher sind, wenn sie auf diesen zugreifen. Wenn zum Beispiel der Server Telnet- oder FTP-Dienste über ein öffentliches Netzwerk zur Verfügung stellt, kann ein Angreifer die Klartext-Benutzernamen und -Passwörter abgreifen, wenn diese über das Netzwerk übertragen werden, und dann diese Account-Informationen zum Zugriff auf den Arbeitsplatzrechner des Remote-Benutzers missbrauchen.

Selbst wenn sichere Protokolle wie z. B. SSH verwendet werden, kann ein Remote-Benutzer anfällig für bestimmte Angriffe sein, wenn ihre Client-Applikationen nicht auf dem neuesten Stand sind. So kann zum Beispiel ein v.1 SSH Client anfällig sein für eine X-Forwarding-Attacke eines böswilligen SSH-Servers. Sobald dieser mit dem Server verbunden ist, kann der Angreifer unbemerkt sämtliche Tastatureingaben und Mausklicks des Benutzers im Netzwerk registrieren. Dieses Problem wurde im v.2 SSH-Protokoll behoben, es liegt jedoch am Benutzer, festzustellen, welche Applikationen solche Anfälligkeiten aufweisen und diese falls nötig auf den neuesten Stand zu bringen.

Abschnitt 2.1, „Sicherheit eines Arbeitsplatzrechners“ beschreibt im Detail, welche Schritte Administratoren und Heimanwender unternehmen sollten, um die Anfälligkeit von Arbeitsplatzrechnern und Heim-PCs einzuschränken.

1.4. Häufige Sicherheitslücken und Angriffe

Tabelle 1.1, „Häufige Sicherheitslücken“ zeigt einige der von Angreifern am häufigsten ausgenutzten Sicherheitslücken und Zugangspunkte, um auf Netzwerkressourcen von Unternehmen zuzugreifen. Der Schlüssel zu diesen häufigen Sicherheitslücken liegt in der Erklärung, wie diese ausgenutzt werden, und wie Administratoren ihr Netzwerk ausreichend gegen solche Angriffe schützen können.

Tabelle 1.1. Häufige Sicherheitslücken

Sicherheitslücke	Beschreibung	Anmerkungen
Null- oder Standardpasswort	Das Leerlassen von administrativen Passwörtern oder das Verwenden von Standardpasswörtern des Herstellers. Dies betrifft häufig Hardware wie Router und Firewalls, jedoch können auch einige Dienste, die unter Linux laufen, standardmäßige Administratorenpasswörter enthalten (Red Hat Enterprise Linux wird jedoch nicht mit diesen ausgeliefert).	Häufig in Verbindung mit Netzwerk-Hardware wie Routern, Firewalls, VPNs und Network-Attached-Storage-Geräten (NAS). Oft in vielen älteren Betriebssystemen, besonders Betriebssysteme, die Dienste kombinieren (wie zum Beispiel UNIX und Windows). Administratoren erzeugen gelegentlich privilegierte Benutzerkonten unter Zeitdruck und lassen Passwörter leer, was einen idealen Einstiegspunkt für böswillige Benutzer bietet, die dieses Benutzerkonto entdecken.
Gemeinsam genutzte Standardschlüssel	Sichere Dienste werden manchmal mit standardmäßigen Sicherheitsschlüsseln für Entwicklung oder zu Evaluierungszwecken ausgeliefert. Werden diese Schlüssel nicht geändert und auf einer Produktionsumgebung im Internet platziert, kann *jeder* Benutzer mit denselben Standardschlüsseln auf diese Ressourcen mit gemeinsam genutzten Schlüsseln und damit auf alle sensiblen Informationen darin zugreifen.	Meistens in Wireless Access Points und vorkonfigurierten sicheren Servergeräten.
IP-Spoofing	Eine sich entfernt befindliche Maschine verhält sich wie ein Knoten im lokalen Netzwerk, findet Schwachstellen auf Ihrem Server und installiert ein Backdoor-Programm oder einen Trojaner, um Kontrolle über Ihre Netzwerkressourcen zu erlangen.	Spoofing ist relativ schwierig, da es vom Angreifer erfordert, dass er TCP/IP Sequenznummern voraussagt, um eine Verbindung zum Zielsystem zu koordinieren. Es sind jedoch verschiedene Tools erhältlich, die dem Cracker bei diesem Angriff helfen können. Beruht auf einem Zielsystem, auf dem Dienste laufen (wie z. B. **`rsh`**, **`telnet`**, FTP und andere), die *Source-basierte* Authentifizierungstechniken verwenden, welche im Vergleich zu PKI oder anderen Formen der Verschlüsselung wie **ssh** oder SSL/TLS nicht empfohlen werden.
Abhören	Das Sammeln von Daten, die zwischen	Diese Art von Angriff funktioniert

	zwei aktiven Knoten auf einem Netzwerk ausgetauscht werden, indem die Verbindung dieser beiden Knoten abgehört wird.	Diese Art von Angriff funktioniert meistens bei Protokollen, bei denen Text unverschlüsselt übertragen wird, wie zum Beispiel Telnet-, FTP- und HTTP-Übertragungen. Angreifer von außerhalb müssen Zugriff auf ein kompromittiertes System in einem LAN haben, um einen derartigen Angriff durchzuführen; üblicherweise hat der Angreifer einen aktiven Angriff (wie zum Beispiel IP-Spoofing oder Man-In-The-Middle) benutzt, um ein System im LAN zu kompromittieren. Präventivmaßnahmen umfassen Dienste mit verschlüsseltem Schlüsselaustausch, Einmal-Passwörtern oder verschlüsselter Authentifizierung, um das Erschnüffeln von Passwörtern zu verhindern; hohe Verschlüsselung während der Übertragung ist ebenfalls ratsam.
Schwachstellen von Diensten	Ein Angreifer findet einen Fehler oder ein Schlupfloch in einem Dienst, der über das Internet läuft. Durch diese Schwachstelle kann der Angreifer das gesamte System und alle Daten darauf sowie weitere Systeme im Netzwerk kompromittieren.	HTTP-basierte Dienste wie CGI sind anfällig für das Ausführen von Remote-Befehlen bis hin zu interaktivem Shell-Zugriff. Auch wenn der HTTP-Dienst als nicht-privilegierter Benutzer wie zum Beispiel "nobody" läuft, können Informationen wie beispielsweise Konfigurationsdateien und Netzwerktabellen gelesen werden, oder der Angreifer kann Denial-of-Service-Attacken starten, die die Ressourcen des Systems beeinträchtigen oder es für weitere Zugriffe durch andere Benutzer unmöglich macht. Dienste können manchmal Schwachstellen haben, die auch nach Entwicklungs- und Testphasen noch existieren; diese Schwachstellen (wie zum Beispiel *Pufferüberläufe*, bei denen Angreifer einen Dienst zum Absturz bringen, indem sie beliebige Werte verwenden, die den Speicherpuffer einer Anwendung füllen und die den Angreifern dann eine interaktive Kommandozeile geben, auf der sie beliebige Befehle ausführen können) können dem Angreifer die

		volle administrative Kontrolle ermöglichen. Administratoren sollten sicherstellen, dass Dienste nicht als Root laufen und sollten aufmerksam nach Patches und Errata-Updates für Applikationen bei Herstellern oder Sicherheitsorganisationen wie CERT und CVE Ausschau halten.
Schwachstellen von Applikationen	Angreifer finden Fehler in Applikationen von Desktops und Arbeitsplatzrechnern (wie z. B. E-Mail-Clients) und führen willkürlich Code aus, implantieren Trojaner für zukünftige Attacken oder bringen Systeme zum Absturz. Noch größerer Schaden kann angerichtet werden, falls der kompromittierte Arbeitsplatzrechner administrative Berechtigungen für den Rest des Netzwerks besitzt.	Arbeitsplatzrechner und Desktops sind im Vergleich zu Servern anfälliger für einen Angriff, da die Benutzer meist nicht die Erfahrung oder das Wissen zur Verhinderung oder Aufdeckung von Einbrüchen haben. Es ist daher zwingend notwendig, diese Benutzer über die Risiken bei der Installation unberechtigter Software oder beim Öffnen vom E-Mail unbekannter Herkunft zu informieren. Es können Schutzeinrichtungen installiert werden, so dass z. B. E-Mail-Software nicht automatisch Anhänge öffnen oder ausführen kann. Zusätzlich dazu kann das automatische Aktualisieren der Software eines Arbeitsplatzrechners über das Red Hat Network oder andere Dienste zur Systemverwaltung die Last einer vielschichtigen Sicherheitsimplementierung etwas mindern.
Denial-of-Service (DoS) Angriffe	Ein Angreifer bzw. eine Gruppe von Angreifern koordiniert eine Attacke auf ein Netzwerk oder auf Serverressourcen eines Unternehmens, bei der unbefugte Pakete an den Zielcomputer (entweder Server, Router oder Arbeitsplatzrechner) gesendet werden. Dies macht die Ressource für berechtigte Benutzer nicht verfügbar.	Der DoS-Fall in den USA, über den am meisten berichtet wurde, trat im Jahr 2000 auf. Eine Reihe hoch frequentierter kommerzieller und Regierungs-Websites wurden vorübergehend außer Gefecht gesetzt durch eine koordinierte Ping-Flood Attacke, bei der mehrere, als *Zombies* agierende, kompromittierte Systeme mit schneller Bandbreitenverbindung benutzt wurden. Quellpakete werden gewöhnlich gefälscht (sowie weiterversendet), was die Suche nach dem wahren Ursprung des Angriffs erschwert. Fortschritte bei Ingress-Filtern (IETF

	rfc2267) durch die Verwendung von **`iptables`** und Network Intrusion Detection Systemen wie zum Beispiel **`snort`** unterstützen Administratoren beim Aufspüren und Verhindern von verteilten DoS-Attacken.

1.5. Sicherheitsaktualisierungen

Wenn Sicherheitslücken in einer Software entdeckt werden, muss die betroffene Software aktualisiert werden, um mögliche Sicherheitsrisiken zu minimieren. Ist die Software Teil eines Pakets einer Red Hat Enterprise Linux Distribution, die derzeit unterstützt wird, liegt es im Interesse von Red Hat, so schnell wie möglich aktualisierte Pakete herauszugeben, die diese Sicherheitslücken schließen. Häufig wird die Mitteilung eines Sicherheitsrisikos von einem Patch begleitet (oder Quell-Code, der den Fehler behebt). Dieses Patch wird dann auf das Red Hat Enterprise Linux Paket angewendet, getestet und als Errata-Update herausgegeben. Enthält die Ankündigung jedoch kein Patch, arbeitet ein Entwickler mit dem Maintainer des Pakets zusammen, um das Problem zu lösen. Wurde das Problem behoben, wird das Paket getestet und als Errata-Update herausgegeben.

Wenn Sie ein Paket verwenden, für das ein Sicherheits-Errata herausgegeben wurde, wird dringend empfohlen, dass Sie die betreffenden Pakete sobald wie möglich aktualisieren, um die Zeit, die Ihr System potenziell angreifbar ist, zu minimieren.

1.5.1. Aktualisieren von Paketen

Wenn Sie Software auf Ihrem System aktualisieren, ist es wichtig, das Update von einer vertrauenswürdigen Quelle herunterzuladen. Ein Angreifer kann leicht eine Version eines Paketes nachbauen (mit der gleichen Versionsnummer des Pakets, das theoretisch das Problem lösen sollte), jedoch ein anderes Sicherheitsrisiko im Paket einbauen, und dieses dann im Internet veröffentlichen. Falls dies geschieht, kann dieses Risiko durch Sicherheitsmaßnahmen wie das Abgleichen der Pakete gegen das ursprüngliche RPM nicht aufgedeckt werden. Es ist daher wichtig, dass Sie RPMs nur von vertrauenswürdigen Quellen wie Red Hat herunterladen und die Signatur des Pakets prüfen, um dessen Integrität sicherzustellen.

Anmerkung

Red Hat Enterprise Linux enthält ein praktisches Symbol in der Menüleiste, das Benachrichtigungen zu verfügbaren Updates anzeigt.

1.5.2. Verifizieren von signierten Paketen

Alle Red Hat Enterprise Linux Pakete sind mit dem Red Hat *GPG*-Schlüssel signiert. GPG steht für GNU Privacy Guard oder GnuPG, ein kostenloses Software-Paket, welches dazu verwendet wird, die Authentizität von Dateien zu gewährleisten. Ein Beispiel: Ein privater Schlüssel (geheimer Schlüssel) hält das Paket verschlossen, wohingegen der öffentliche Schlüssel das Paket verifiziert und freischaltet. Falls der von Red Hat Enterprise Linux ausgegebene öffentliche Schlüssel während der RPM-Verifizierung nicht mit dem privaten Schlüssel übereinstimmt, kann dies bedeuten, dass das Paket in irgendeiner Form verändert wurde und daher nicht vertrauenswürdig ist.

Das RPM-Dienstprogramm in Red Hat Enterprise Linux 6 versucht automatisch, die GPG-Signatur eines RPM-Paketes vor der Installation zu verifizieren. Ist der Red Hat GPG-Schlüssel nicht installiert, sollten

Sie diesen von einer sicheren, statischen Quelle wie einer Red Hat Enterprise Linux Installations-CD oder -DVD installieren.

Angenommen die CD oder DVD ist in **/mnt/cdrom** eingehängt, können Sie den folgenden Befehl zum Importieren des Schlüssels in den *Schlüsselbund* (engl. "Keyring", eine Datenbank bestehend aus vertrauenswürdigen Schlüsseln auf dem System) verwenden.

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Um eine Liste aller installierten Schlüssel für die RPM-Verifikation anzuzeigen, führen Sie folgenden Befehl aus:

```
rpm -qa gpg-pubkey*
```

Die Ausgabe sollte etwa folgendermaßen aussehen:

```
gpg-pubkey-db42a60e-37ea5438
```

Um Details über einen bestimmten Schlüssel anzuzeigen, verwenden Sie den Befehl **rpm -qi**, gefolgt von der Ausgabe des vorherigen Befehls, in diesem Beispiel also:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

Es ist von größter Wichtigkeit, dass Sie die Signatur der RPM-Dateien verifizieren, bevor Sie diese installieren. Dieser Schritt gewährleistet, dass die RPMs der Pakete nicht verändert wurden. Um alle heruntergeladenen Pakete gleichzeitig zu prüfen, geben Sie folgenden Befehl ein:

```
rpm -K /tmp/updates/*.rpm
```

Für jedes einzelne Paket erhalten Sie im Falle einer erfolgreichen Verifikation die Ausgabe **gpg OK**. Ist dies nicht der Fall, überprüfen Sie, ob Sie den richtigen öffentlichen Schlüssel von Red Hat verwenden und verifizieren Sie die Quelle des Inhalts. Pakete, welche die GPG-Verifizierung nicht bestehen, sollten nicht installiert werden, da sie möglicherweise von Dritten verändert wurden.

Nachdem der GPG-Schlüssel verifiziert und alle Pakete im Zusammenhang mit der Errata-Meldung heruntergeladen wurden, können Sie diese als Root an einem Shell-Prompt installieren.

1.5.3. Installieren von signierten Paketen

Die Installation für die meisten Pakete kann auf sichere Weise durch den folgenden Befehl durchgeführt werden (Kernel-Pakete ausgenommen):

```
rpm -Uvh /tmp/updates/*.rpm
```

Für Kernel-Pakete sollten Sie den folgenden Befehl verwenden:

```
rpm -ivh /tmp/updates/<kernel-package>
```

Ersetzen Sie ***<kernel-package>*** im obigen Beispiel durch den Namen des Kernel-RPM.

Nachdem der Rechner unter Verwendung des neuen Kernels sicher neu gestartet ist, kann der alte Kernel mit dem folgenden Befehl entfernt werden:

```
rpm -e <old-kernel-package>
```

Ersetzen Sie ***<old-kernel-package>*** im obigen Beispiel durch den Namen des älteren Kernel-RPM.

Anmerkung

Es ist nicht unbedingt erforderlich, dass der alte Kernel entfernt wird. Der standardmäßige Bootloader, GRUB, erlaubt mehrere installierte Kernel, aus denen dann während des Boot-Vorgangs im Menü einer ausgewählt werden kann.

Wichtig

Vergewissern Sie sich, dass Sie vor der Installation von Sicherheits-Errata jegliche speziellen Anweisungen in der Errata-Meldung lesen und diese entsprechend befolgen. Siehe Abschnitt 1.5.4, „Anwenden der Änderungen“ für allgemeine Anweisungen zum Anwenden von Änderungen durch ein Errata-Update.

1.5.4. Anwenden der Änderungen

Nachdem Sie die Sicherheits-Errata und Aktualisierungen heruntergeladen und installiert haben, ist es wichtig, die ältere Software nicht mehr einzusetzen, sondern stattdessen die neue Software zu verwenden. Die Vorgehensweise hängt von der Art der Software ab, die aktualisiert wurde. Die folgende Liste stellt die allgemeinen Kategorien der Software dar und gibt Anweisungen für das Verwenden der aktualisierten Versionen nach einer Paketaktualisierung.

Anmerkung

Im Allgemeinen ist ein Neustart der beste Weg um sicherzustellen, dass die aktuellste Version eines Software-Pakets verwendet wird, allerdings ist dies für den Systemadministrator nicht immer machbar.

Applikationen

Bei User-Space-Applikationen handelt es sich um alle Programme, die durch einen Systembenutzer gestartet werden können. Für gewöhnlich laufen diese Anwendungen nur, wenn ein Benutzer, ein Skript oder ein automatisiertes Dienstprogramm diese startet, und sie werden in der Regel nicht für längere Zeit ausgeführt.

Wird solch eine User-Space-Applikation aktualisiert, stoppen Sie alle Instanzen dieser Anwendung auf dem System und starten Sie das Programm erneut, um die aktualisierte Version zu verwenden.

Kernel

Der Kernel ist die Kern-Software-Komponente für das Red Hat Enterprise Linux Betriebssystem. Er verwaltet den Zugriff auf den Speicher, den Prozessor und auf Peripheriegeräte, und plant sämtliche Aufgaben.

Aufgrund seiner zentralen Rolle kann der Kernel nur durch ein Herunterfahren des Computers neu gestartet werden. Daher kann eine aktualisierte Version des Kernels erst verwendet

werden, wenn das System neu gestartet wird.

Gemeinsam verwendete Bibliotheken

Gemeinsam verwendete Bibliotheken sind Einheiten von Code, wie z. B. **glibc**, die von einer Reihe von Applikationen und Software-Programmen gemeinsam verwendet werden. Applikationen, die gemeinsam verwendete Bibliotheken nutzen, laden normalerweise den gemeinsamen Code beim Starten der Anwendungen, so dass alle Applikationen, die die aktualisierte Bibliothek verwenden, neu gestartet werden müssen.

Um festzustellen, welche laufenden Applikationen mit einer bestimmten Bibliothek verknüpft sind, verwenden Sie den Befehl **lsof** wie im folgenden Beispiel:

```
lsof /lib/libwrap.so*
```

Dieser Befehl gibt eine Liste aller laufenden Programme aus, die TCP-Wrappers für die Host-Zugangskontrolle verwenden. Alle aufgelisteten Programme müssen angehalten und neu gestartet werden, wenn das **tcp_wrappers**-Paket aktualisiert wird.

SysV-Dienste

SysV-Dienste sind persistente Server-Programme, die während des Bootens gestartet werden. Beispiele für SysV-Dienste sind **sshd**, **vsftpd** und **xinetd**.

Da diese Programme normalerweise im Speicher verbleiben, solange der Rechner läuft, muss jeder aktualisierte SysV-Dienst nach der Aktualisierung des Pakets angehalten und neu gestartet werden. Dies kann über das **Tool zur Dienstkonfiguration** oder durch Anmelden als Root via Shell-Prompt und Ausführen des Befehls **/sbin/service** erreicht werden, wie im folgenden Beispiel veranschaulicht:

```
/sbin/service <service-name> restart
```

Ersetzen Sie im obigen Beispiel ***<service-name>*** durch den Namen des Dienstes, wie z. B. **sshd**.

xinetd-Dienste

Dienste, die vom Super-Dienst **xinetd** gesteuert werden, werden nur ausgeführt, wenn eine aktive Verbindung vorliegt. Von **xinetd** gesteuert werden z. B. die Telnet, IMAP und POP3-Dienste.

Da **xinetd** jedesmal neue Instanzen dieser Dienste startet, wenn eine neue Anfrage empfangen wird, werden die Verbindungen, die nach einer Aktualisierung entstehen, durch die aktualisierte Software gesteuert. Bestehen jedoch zu dem Zeitpunkt, an dem von **xinetd** verwaltete Dienste aktualisiert werden, aktive Verbindungen, so werden diese noch von der älteren Version der Software bedient.

Um ältere Instanzen eines bestimmten **xinetd**-Dienstes zu stoppen, aktualisieren Sie das Paket für den Dienst und stoppen Sie anschließend alle aktuell laufenden Prozesse. Mit dem Befehl **ps** können Sie feststellen, welche Prozesse laufen. Geben Sie dann den Befehl **kill** oder **killall** ein, um alle aktuellen Instanzen dieses Dienstes zu stoppen.

Wenn zum Beispiel Sicherheits-Errata für die **imap**-Pakete herausgegeben werden, aktualisieren Sie die Pakete und geben Sie danach folgenden Befehl als Root ein:

```
ps aux | grep imap
```

Dieser Befehl gibt alle aktiven IMAP-Sitzungen aus. Einzelne Sitzungen können dann mithilfe des folgenden Befehls beendet werden:

```
kill <PID>
```

Falls das Beenden der Sitzung damit fehlschlägt, verwenden Sie stattdessen folgenden Befehl:

```
kill -9 <PID>
```

Ersetzen Sie im obigen Beispiel ***<PID>*** durch die Prozess-Identifikationsnummer (zu finden in der zweiten Spalte des **ps**-Befehls) der fraglichen IMAP-Sitzung.

Um alle aktiven IMAP-Sitzungen zu beenden, geben Sie den folgenden Befehl ein:

```
killall imapd
```

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html

[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/

[4] http://www.fudzilla.com/content/view/7847/1/

[5] http://www.internetworldstats.com/stats.htm

[6] http://www.cert.org

[7] http://www.cert.org/stats/cert_stats.html

[8] http://news.cnet.com/Computer-crime-costs-67-billion,-FBI-says/2100-7349_3-6028946.html

[9] http://www.cio.com/article/504837/Why_Security_Matters_Now

[10] http://www.sans.org/resources/errors.php

Kapitel 2. Sichern Ihres Netzwerks

2.1. Sicherheit eines Arbeitsplatzrechners

Die Sicherung einer Linux-Umgebung beginnt beim Arbeitsplatzrechner. Ungeachtet dessen, ob Sie Ihren persönlichen Rechner oder ein Firmensystem sichern, beginnt eine vernünftige Sicherheitsrichtlinie mit dem einzelnen Computer. Im Endeffekt ist ein Computernetzwerk nur so sicher wie das schwächste Glied.

2.1.1. Beurteilung der Arbeitsplatzrechner-Sicherheit

Wenn Sie die Sicherheit eines Red Hat Enterprise Linux Arbeitsplatzrechners auswerten, sollten Sie Folgendes beachten:

- *BIOS und Bootloader-Sicherheit* — Kann ein unbefugter Benutzer physisch auf den Rechner zugreifen und in den Einzelbenutzer- oder Rettungsmodus booten, ohne dass nach einem Passwort gefragt wird?
- *Passwortsicherheit* — Wie sicher sind die Passwörter für die Benutzeraccounts auf dem Rechner?
- *Administrative Kontrolle* — Wer hat alles einen Account auf dem System, und wie viel administrative Kontrolle wird diesen Accounts gewährt?
- *Verfügbare Netzwerkdienste* — Welche Dienste horchen auf dem Netzwerk auf Anfragen, und sollten diese überhaupt aktiv sein?
- *Persönliche Firewalls* — Welche Art von Firewall, wenn überhaupt, ist nötig?
- *Kommunikationstools mit erweiterter Sicherheit* — Welche Tools sollten zur Kommunikation zwischen Arbeitsplatzrechnern verwendet werden, und welche sollten vermieden werden?

2.1.2. BIOS und Bootloader-Sicherheit

Passwortschutz für das BIOS (oder BIOS-Äquivalent) und den Bootloader kann unbefugte Benutzer, die physischen Zugang zu Ihren Systemen erlangen, davon abhalten, externe Medien zu booten oder sich durch den Einzelbenutzermodus als Root anzumelden. Die Sicherheitsmaßnahmen, mithilfe derer Sie sich vor solchen Attacken schützen sollten, hängen zum einen von den Informationen ab, die auf dem Arbeitsplatzrechner gespeichert sind, und zum anderen vom Standort des Rechners.

Wenn zum Beispiel ein Computer auf einer Messe verwendet wird und keine sensiblen Daten enthält, ist es nicht unbedingt wichtig, solche Attacken zu verhindern. Wenn jedoch ein Laptop eines Mitarbeiters mit privaten, nicht-passwortgeschützten SSH-Schlüsseln zum Firmennetzwerk auf der gleichen Messe unbeaufsichtigt gelassen wird, kann dies zu einem bedeutenden Sicherheitsbruch führen, der Auswirkungen auf das gesamte Unternehmen haben kann.

Wenn sich der Rechner dagegen an einem Ort befindet, zu dem nur befugte oder vertrauenswürdige Personen Zugang haben, ist das Sichern des BIOS oder des Bootloaders nicht unbedingt notwendig.

2.1.2.1. BIOS-Passwörter

Es gibt zwei Hauptgründe für den Schutz des BIOS eines Computers durch Passwörter [11]:

1. *Änderungen an den BIOS-Einstellungen verhindern* — Hat ein Eindringling Zugang zum BIOS, kann dieser den Bootvorgang von einer Diskette oder einer CD-ROM festlegen. Dies ermöglicht dann, in den Rettungsmodus oder Einzelbenutzermodus zu gelangen und von hier aus schädliche Prozesse auf dem System zu starten oder sensible Daten zu kopieren.
2. *System-Boot verhindern* — Einige BIOS erlauben Ihnen, den Bootvorgang selbst mit einem Passwort zu schützen. Ist dies aktiviert, muss ein Passwort eingegeben werden, bevor das BIOS

den Bootloader startet.

Da die Methoden für das Einstellen von BIOS-Passwörtern sich von Hersteller zu Hersteller unterscheiden, lesen Sie bitte das Handbuch Ihres Computers für weitere Informationen.

Sollten Sie das BIOS-Passwort vergessen, kann es oft entweder mit Jumpern im Motherboard oder durch das Entfernen der CMOS-Batterie zurückgesetzt werden. Daher ist es sinnvoll, wenn möglich das Computergehäuse abzuschließen. Bevor Sie jedoch versuchen, die CMOS-Batterie zu entfernen, sollten Sie das Handbuch Ihres Computers oder Motherboards lesen.

2.1.2.1.1. Sicherung von nicht-x86-Plattformen

Andere Architekturen verwenden verschiedene Programme zum Ausführen von Low-Level-Aufgaben, die mit denen des BIOS auf einem x86-System äquivalent sind. So verwenden zum Beispiel Intel® Itanium™-basierte Computer die *Extensible Firmware Interface* (*EFI*) Shell.

Anweisungen für den Passwortschutz von BIOS-ähnlichen Programmen auf anderen Architekturen finden Sie in den Handbüchern des Herstellers.

2.1.2.2. Bootloader-Passwörter

Es gibt die folgenden wesentlichen Gründe für den Schutz eines Linux-Bootloaders:

1. *Zugang zum Einzelbenutzermodus verhindern* — Wenn Angreifer in den Einzelbenutzermodus booten können, werden diese automatisch zu Root-Benutzern, ohne nach dem Root-Passwort gefragt zu werden.
2. *Zugang zur GRUB-Konsole verhindern* — Wenn der Rechner GRUB als Bootloader verwendet, kann ein Angreifer die GRUB-Editor-Schnittstelle verwenden, um die Konfiguration zu ändern oder Informationen mithilfe des **`cat`**-Befehls zu sammeln.
3. *Zugang zu unsicheren Betriebssystemen verhindern* — Haben Sie ein Dual-Boot-System, kann ein Angreifer während des Bootens ein Betriebssystem wie zum Beispiel DOS auswählen, das Zugangskontrollen und Dateiberechtigungen ignoriert.

Red Hat Enterprise Linux 6 wird mit dem GRUB-Bootloader für die x86-Plattform ausgeliefert. Detaillierte Informationen zu GRUB finden Sie im Red Hat Installationshandbuch.

2.1.2.2.1. Passwortschutz für GRUB

Sie können GRUB konfigurieren, um die ersten beiden, in Abschnitt 2.1.2.2, „Bootloader-Passwörter“ angesprochenen Probleme zu vermeiden, indem Sie eine Passwortdirektive zur Konfigurationsdatei hinzufügen. Hierfür legen Sie erst ein sicheres Passwort fest, öffnen dann einen Shell-Prompt, melden sich als Root an und geben Folgendes ein:

```
/sbin/grub-md5-crypt
```

Wenn Sie dazu aufgefordert werden, geben Sie das GRUB-Passwort ein und drücken anschließend **`Enter`**. Daraufhin wird ein MD5-Hash des Passworts ausgegeben.

Bearbeiten Sie als Nächstes die GRUB-Konfigurationsdatei **`/boot/grub/grub.conf`**. Öffnen Sie die Datei und fügen Sie die nachfolgende Zeile unterhalb der **`timeout`**-Zeile im Hauptabschnitt des Dokuments ein:

```
password --md5 <password-hash>
```

Ersetzen Sie ***<password-hash>*** durch den Wert, der von **`/sbin/grub-md5-crypt`** [12] ausgegeben wurde.

Wenn Sie das nächste Mal Ihr System booten, verhindert das GRUB-Menü den Zugriff auf den Editor oder die Befehlszeilen-Schnittstelle, bis Sie **p** drücken und dann das GRUB-Passwort eingeben.

Diese Lösung hält jedoch Angreifer nicht davon ab, in einer Dual-Boot-Umgebung in ein unsicheres Betriebssystem zu booten. Hierfür müssen Sie einen anderen Teil der Datei **/boot/grub/grub.conf** bearbeiten.

Suchen Sie die **title**-Zeile des Betriebssystems, das sie absichern möchten, und fügen Sie direkt darunter eine Zeile mit der **lock**-Direktive ein.

Für ein DOS-System sollte der Absatz etwa wie folgt beginnen:

```
title DOS lock
```

Warnung

Es muss eine **password**-Zeile im Hauptabschnitt der **/boot/grub/grub.conf**-Datei vorhanden sein, damit diese Methode funktionieren kann. Andernfalls kann ein Angreifer auf den GRUB-Editor zugreifen und die lock-Zeile entfernen.

Wenn Sie für einen bestimmten Kernel oder ein Betriebssystem ein anderes Passwort festlegen möchten, fügen Sie eine **lock**-Zeile gefolgt von einer Passwortzeile in den Absatz ein.

Jeder Absatz, den Sie mit einem eindeutigen Passwort schützen möchten, sollte mit einer Zeile ähnlich dem folgenden Beispiel beginnen:

```
title DOS lock password --md5 <password-hash>
```

2.1.3. Passwortsicherheit

Passwörter werden in Red Hat Enterprise Linux als Hauptmethode zur Überprüfung der Benutzeridentität eingesetzt. Aus diesem Grund ist die Passwortsicherheit sehr wichtig für den Schutz des Benutzers, des Arbeitsplatzrechners und des Netzwerks.

Aus Sicherheitsgründen konfiguriert das Installationsprogramm das System zur Verwendung von *Secure Hash Algorithm 512* (*SHA512*) und Shadow-Passwörtern. Es wird dringend empfohlen, diese Einstellungen nicht zu verändern.

Wenn Sie die Shadow-Passwörter während der Installation deaktivieren, werden alle Passwörter als unidirektionaler Hash in der allgemein lesbaren **/etc/passwd**-Datei gespeichert, wodurch das System potenziell für Angriffe verwundbar wird, bei denen Passwörter offline geknackt werden. Erlangt ein Angreifer als regulärer Benutzer Zugriff auf das System, kann er die **/etc/passwd**-Datei auf seinen eigenen Rechner kopieren und diverse Programme zum Knacken von Passwörtern darüber laufen lassen. Befindet sich ein unsicheres Passwort in der Datei, ist es nur eine Frage der Zeit, bis es von diesen Programmen geknackt wird.

Shadow-Passwörter machen diese Art von Angriff unmöglich, da die Passwort-Hashes in der Datei **/etc/shadow** gespeichert werden, die nur vom Root-Benutzer gelesen werden kann.

Dies zwingt einen möglichen Angreifer, Passwörter auf dem Rechner von Remote aus über einen Netzwerkdienst wie zum Beispiel SSH oder FTP zu knacken. Diese Art von Angriff ist wesentlich langsamer und hinterlässt offensichtliche Spuren, da Hunderte von gescheiterten Anmeldeversuchen in

Systemdateien aufgezeichnet werden. Wenn der Angreifer jedoch eine Attacke mitten in der Nacht startet und Sie schwache Passwörter auf dem System haben, erlangt der Angreifer eventuell Zugang noch vor Morgengrauen und konnte seine Spuren in den Protokolldateien bereits verwischen.

Ein weiteres Problem über die Überlegungen zu Format und Speicherung hinaus, ist der Inhalt. Das wichtigste, was ein Benutzer tun kann, um seinen Account gegen eine Passwortattacke zu schützen, ist das Erstellen eines sicheren Passworts.

2.1.3.1. Erstellen sicherer Passwörter

Beim Erstellen von Passwörtern ist es sinnvoll, die folgenden Richtlinien zu befolgen:

- *Verwenden Sie nicht nur Wörter oder Zahlen* — Sie sollten für ein Passwort nicht ausschließlich Wörter oder ausschließlich Zahlen verwenden.

  Hier einige Beispiele für unsichere Passwörter:

  - 8675309
  - juan
  - hackme

- *Verwenden Sie keine erkennbaren Wörter* — Wörter wie Namen, im Wörterbuch stehende Wörter oder Begriffe aus Fernsehsendungen oder Romanen sollten vermieden werden, auch wenn diese am Ende mit Zahlen versehen werden.

  Hier einige Beispiele für unsichere Passwörter:

  - john1
  - DS-9
  - mentat123

- *Verwenden Sie keine Wörter in anderen Sprachen* — Programme zum Knacken von Passwörtern prüfen oft anhand von Wortlisten, die Wörterbücher in anderen Sprachen umfassen. Sich für sichere Passwörter auf Fremdsprachen zu verlassen ist daher häufig wenig hilfreich.

  Hier einige Beispiele für unsichere Passwörter:

  - cheguevara
  - bienvenido1
  - 1dummkopf

- *Verwenden Sie keine Hacker-Begriffe* — Glauben Sie nicht, dass Sie auf der sicheren Seite sind, wenn Sie Hacker-Begriffe — auch l337 (LEET) genannt — für Ihre Passwörter verwenden. Viele Wortlisten enthalten LEET-Begriffe.

  Hier einige Beispiele für unsichere Passwörter:

  - H4X0R
  - 1337

- *Verwenden Sie keine persönlichen Informationen* — Vermeiden Sie die Verwendung von persönlichen Informationen in Ihren Passwörtern. Wenn der Angreifer Sie kennt, kann er Ihr Passwort leichter herausfinden. Sehen Sie nachfolgend eine Liste mit zu vermeidenden Informationen beim Erstellen eines Passworts:

  Hier einige Beispiele für unsichere Passwörter:

  - Ihren Namen
  - Den Namen von Haustieren
  - Die Namen von Familienmitgliedern
  - Jegliche Geburtstage
  - Ihre Telefonnummer oder Postleitzahl

- *Drehen Sie keine erkennbaren Wörter um* — Gute Passwortprogramme überprüfen

gemeinsprachliche Wörter auch rückwärts, das Invertieren von schlechten Passwörtern machen diese also nicht sicherer.

Hier einige Beispiele für unsichere Passwörter:

- R0X4H
- nauj
- 9-DS

- *Schreiben Sie sich Ihr Passwort nicht auf* — Bewahren Sie Ihr Passwort niemals auf Papier auf. Es ist wesentlich sicherer, sich das Passwort zu merken.
- *Verwenden Sie nie das gleiche Passwort für alle Ihre Rechner* — Es ist wichtig, dass Sie separate Passwörter für jeden Recher erstellen. So sind nicht alle Rechner auf einen Schlag betroffen, falls ein System einem Angriff zum Opfer fällt.

Die folgenden Richtlinien helfen Ihnen dabei, ein sicheres Passwort zu erstellen:

- *Das Passwort sollte mindestens acht Zeichen enthalten* — Je länger das Passwort, desto besser. Wenn Sie MD5-Passwörter verwenden, sollten diese 15 Zeichen oder mehr enthalten. DES-Passwörter sollten die maximale Länge nutzen (acht Zeichen).
- *Mischen Sie Groß- und Kleinbuchstaben* — In Red Hat Enterprise Linux wird Groß- und Kleinschreibung unterschieden, mischen Sie daher Groß- und Kleinbuchstaben, um die Sicherheit des Passworts zu erhöhen.
- *Mischen Sie Buchstaben und Zahlen* — Das Hinzufügen von Zahlen, insbesondere in der Mitte des Passwortes (nicht nur am Anfang oder Ende), verstärkt die Sicherheit des Passworts.
- *Verwenden Sie Sonderzeichen* — Nicht-alphanumerische Zeichen wie z. B. &, $ und > können die Sicherheit des Passworts signifikant erhöhen (nicht möglich für DES-Passwörter).
- *Wählen Sie ein Passwort, das Sie sich leicht merken können* — selbst das beste Passwort hilft Ihnen nicht weiter, wenn Sie sich nicht daran erinnern können. Verwenden Sie daher Akronyme oder andere mnemonische Techniken, um sich das Passwort zu merken.

Durch all diese Regeln erscheint es schwierig, ein Passwort zu erstellen, das all die Kriterien für sichere Passwörter erfüllt und gleichzeitig die Charakteristiken von schlechten Passwörtern vermeidet. Glücklicherweise gibt es einige einfache Schritte, mit deren Hilfe Sie ein leicht zu merkendes, sicheres Passwort generieren können.

2.1.3.1.1. Methode zur Erstellung sicherer Passwörter

Es gibt viele verschiedene Methoden, sichere Passwörter zu erstellen. Eine der beliebtesten Methoden verwendet Akronyme. Zum Beispiel:

- Überlegen Sie sich einen leicht zu merkenden Satz, wie zum Beispiel:

  "over the river and through the woods, to grandmother's house we go."
- Verwandeln Sie dies als Nächstes in ein Akronym (einschließlich der Satzzeichen).

  `otrattw,tghwg.`
- Machen Sie das Passwort komplexer, indem Sie Buchstaben durch Zahlen und Sonderzeichen austauschen. Ersetzen Sie zum Beispiel **`t`** durch **`7`** und **`a`** durch das at-Symbol (`@`):

  `o7r@77w,7ghwg.`
- Machen Sie es noch komplexer, indem Sie mindestens einen Buchstaben groß schreiben, zum Beispiel **`H`**.

  `o7r@77w,7gHwg.`
- *Und bitte verwenden Sie nicht unser Beispielpasswort für Ihre Systeme.*

Das Erstellen sicherer Passwörter ist von größter Wichtigkeit, genauso wichtig ist jedoch die richtige

Verwaltung der Passwörter, insbesondere für Systemadministratoren in größeren Unternehmen. Im nächsten Abschnitt werden Verfahren für das Erstellen und Verwalten von Benutzerpasswörtern innerhalb eines Unternehmens beschrieben.

2.1.3.2. Erstellen von Benutzerpasswörtern innerhalb eines Unternehmens

Wenn es eine große Anzahl von Benutzern in einem Unternehmen gibt, haben Systemadministratoren zwei grundlegende Möglichkeiten, um die Verwendung sicherer Passwörter zu erzwingen. Sie können entweder selbst Passwörter für die Benutzer erstellen, oder aber Benutzer ihre eigenen Passwörter erstellen lassen und dabei deren Qualität überprüfen.

Das Erstellen der Passwörter für den Benutzer stellt sicher, dass die Passwörter sicher sind, kann aber schnell zu einer ausufernden Arbeit werden, wenn das Unternehmen wächst. Außerdem erhöht dies das Risiko, dass die Benutzer ihre Passwörter aufschreiben.

Aus diesen Gründen ziehen es die meisten Systemadministratoren vor, dass die Benutzer ihre eigenen Passwörter erstellen, diese jedoch auf ihre Sicherheit prüfen und in einigen Fällen Benutzer durch die Passwortalterung dazu zu zwingen, ihre Passwörter in regelmäßigen Abständen zu ändern.

2.1.3.2.1. Erzwingen sicherer Passwörter

Um das Netzwerk vor Eindringlingen zu schützen, sollten Systemadministratoren sicherstellen, dass die im Unternehmen verwendeten Passwörter sicher sind. Wenn Benutzer aufgefordert werden, ihre eigenen Passwörter zu erstellen oder zu ändern, können sie dies über die Befehlszeilenapplikation **`passwd`** tun, die *Pluggable Authentication Manager* (*PAM*) unterstützt und daher prüft, ob ein Passwort zu kurz oder anderweitig zu unsicher ist. Diese Prüfung erfolgt mit dem **`pam_cracklib.so`**-PAM-Modul. Da PAM anpassbar ist, ist es möglich, weitere Passwort-Integritätsprüfer hinzuzufügen wie z. B. **`pam_passwdqc`** (erhältlich unter http://www.openwall.com/passwdqc/) oder ein neues Modul zu schreiben. Eine Liste erhältlicher PAM-Module finden Sie unter http://www.kernel.org/pub/linux/libs/pam/modules.html. Weitere Informationen über PAM finden Sie unter *Managing Single Sign-On and Smart Cards*.

Die Passwortprüfung zum Erstellungszeitpunkt erkennt schlechte Passwörter jedoch nicht so effektiv wie ein Programm zum Knacken von Passwörtern.

Es gibt eine Vielzahl an Passwort-Cracking-Programmen, die unter Red Hat Enterprise Linux laufen, jedoch nicht mit dem Betriebssystem ausgeliefert werden. Nachfolgend finden Sie eine kurze Liste der beliebtesten Passwort-Cracking-Programme:

- ***John The Ripper*** — Ein schnelles und flexibles Passwort-Cracking-Programm. Es ermöglicht die Verwendung mehrerer Wortlisten und ist fähig zum Brute-Force Passwort-Cracking. Es ist unter http://www.openwall.com/john/ erhältlich.
- ***Crack*** — Die vielleicht bekannteste Passwort-Cracking-Software. **Crack** ist ebenfalls sehr schnell, jedoch nicht so einfach zu verwenden wie **John The Ripper**. Es ist unter http://www.crypticide.com/alecm/security/crack/c50-faq.html erhältlich.
- ***Slurpie*** — **Slurpie** funktioniert ähnlich wie **John The Ripper** und **Crack**, ist jedoch darauf ausgelegt, auf mehreren Computern gleichzeitig zu laufen und ermöglicht so einen verteilten Passwort-Cracking-Angriff. Es ist erhältlich unter http://www.ussrback.com/distributed.htm, zusammen mit einer Reihe anderer Tools zur Bewertung der Sicherheit bei verteilten Passwort-Cracking-Angriffen.

Warnung

Bitte holen Sie sich stets eine schriftliche Genehmigung ein, bevor Sie Passwörter innerhalb eines Unternehmens zu knacken versuchen.

2.1.3.2.2. Passphrasen

Passphrasen und Passwörter bilden die Eckpfeiler der Sicherheit in den meisten heutigen Systemen. Leider gehören Technologien wie biometrische Daten oder Zwei-Wege-Authentifikation in den meisten Systemen noch nicht zum Standard. Wenn Passwörter zur Sicherung eines Systems verwendet werden müssen, sollten stattdessen Passphrasen in Erwägung gezogen werden. Passphrasen sind länger als Passwörter und bieten eine höhere Sicherheit selbst im Vergleich mit Passwörtern, die Sonderzeichen und Ziffern enthalten.

2.1.3.2.3. Passwortalterung

Passwortalterung ist eine weitere Methode, die von Systemadministratoren verwendet wird, um unsicheren Passwörtern in einem Unternehmen vorzubeugen. Passwortalterung bedeutet, dass ein Benutzer nach einer bestimmten Zeit (gewöhnlich 90 Tage) aufgefordert wird, ein neues Passwort festzulegen. Wird der Benutzer regelmäßig zur Änderung seines Passworts gezwungen, ist somit selbst ein geknacktes Passwort dem Angreifer nur für eine begrenzte Zeit nützlich. Der Nachteil der Passwortalterung ist jedoch, dass Benutzer eher dazu neigen, sich die Passwörter aufzuschreiben.

Es gibt zwei Programme für das Festlegen der Passwortalterung unter Red Hat Enterprise Linux: den Befehl **chage** oder die grafische **Benutzerverwaltung** (**system-config-users**).

Die Option **-M** des **chage**-Befehls legt die maximale Anzahl von Tagen fest, für die das Passwort gültig ist. Wenn Sie zum Beispiel festlegen wollen, dass ein Benutzerpasswort nach 90 Tagen ungültig wird, geben Sie den folgenden Befehl ein:

```
chage -M 90 <username>
```

Ersetzen Sie im oben genannten Befehl ***<username>*** mit dem Namen des Benutzers. Wenn Sie nicht möchten, dass das Passwort ungültig wird, verwenden Sie den Wert **99999** nach der Option **-M** (dies entspricht etwas mehr als 273 Jahren).

Sie können den Befehl **chage** auch im interaktiven Modus verwenden, um mehrere Details der Passwortalterung und des Benutzerkontos zu ändern. Verwenden Sie folgenden Befehl für den interaktiven Modus:

```
chage <username>
```

Nachfolgend sehen Sie eine interaktive Beispielsession mit diesem Befehl:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@myServer ~]#
```

Werfen Sie einen Blick auf die Handbuchseite für weitere Informationen über verfügbare Optionen.

Sie können auch die grafische Applikation zur **Benutzerverwaltung** nutzen, um wie nachfolgend beschrieben Richtlinien zur Passwortalterung festzulegen. Beachten Sie, dass Sie zur Durchführung dieses Verfahrens über Administratorrechte verfügen müssen.

1. Klicken Sie im **System**-Menü auf der oberen Menüleiste auf **Administration** und anschließend auf **Benutzer und Gruppen**, um die Benutzerverwaltung anzuzeigen. Alternativ können Sie dazu auch den Befehl **system-config-users** an einem Shell-Prompt eingeben.
2. Klicken Sie auf den **Benutzer**-Reiter und wählen Sie den gewünschten Benutzer aus der Liste aus.
3. Klicken Sie auf **Eigenschaften** in der Werkzeugleiste, um das Dialogfeld mit den Benutzereigenschaften anzuzeigen (oder wählen Sie **Eigenschaften** aus dem **Datei**-Menü).
4. Klicken Sie auf den **Passwort-Info**-Reiter und markieren Sie das Auswahlkästchen **Ablauf des Passworts aktivieren**.
5. Geben Sie im Feld **Verbleibende Tage bis zur Änderung** den gewünschten Wert ein und klicken Sie anschließend auf **OK**.

Abbildung 2.1. Angeben der Optionen zur Passwortalterung

2.1.4. Administrative Kontrolle

Bei der Verwaltung eines Heimcomputers muss der Benutzer einige Aufgaben als Root-Benutzer

durchführen, oder unter Verwendung eines *setuid*-Programms wie **sudo** oder **su**. Ein setuid-Programm ist ein Programm, das mit der Benutzer-ID (*UID*) des Besitzers dieses Programms ausgeführt wird, statt mit der Benutzer-ID desjenigen Benutzers, der dieses Programm ausführt. Solche Programme sind durch ein **s** im Besitzerabschnitt eines ausführlichen Listings gekennzeichnet, wie im folgenden Beispiel veranschaulicht:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```

Anmerkung

Das **s** kann ein Groß- oder Kleinbuchstabe sein. Falls es ein Großbuchstabe ist, bedeutet dies, dass das darunterliegende Berechtigungs-Bit nicht gesetzt ist.

Systemadministratoren eines Unternehmens dagegen müssen festlegen, in welchem Umfang die Benutzer im Unternehmen administrative Kontrolle über ihre Computer erhalten dürfen. Mithilfe des PAM-Moduls namens **pam_console.so** können einige Vorgänge, die normalerweise nur dem Root-Benutzer erlaubt sind, wie z. B. das Neustarten und Einhängen von Wechseldatenträgern, dem ersten Benutzer erlaubt werden, der sich an der physischen Konsole anmeldet (siehe auch *Managing Single Sign-On and Smart Cards* für weitere Informationen über das **pam_console.so**-Modul). Andere wichtige Systemadministrationsaufgaben wie das Ändern von Netzwerkeinstellungen, Konfigurieren einer neuen Maus oder das Einhängen von Netzwerkgeräten sind jedoch ohne Administratorrechte nicht möglich, weshalb Systemadministratoren entscheiden müssen, in welchem Umfang die Benutzer in ihrem Netzwerk administrative Kontrolle erhalten sollen.

2.1.4.1. Gewähren von Root-Zugriff

Sind die Benutzer innerhalb eines Unternehmens vertrauenswürdig und computerversiert, ist das Vergeben von Root-Berechtigungen unter Umständen sinnvoll. Root-Zugang zu erlauben bedeutet, dass kleinere Probleme wie das Hinzufügen von Geräten oder das Konfigurieren von Netzwerkschnittstellen von den einzelnen Benutzern selbst durchgeführt werden kann und somit Systemadministratoren mehr Zeit für Netzwerksicherheit und andere, wichtige Aufgaben haben.

Andererseits kann das Vergeben von Root-Rechten an Einzelbenutzer zu folgenden Problemen führen:

- *Fehlkonfiguration des Rechners* — Benutzer mit Root-Rechten können ihre Computer unter Umständen falsch konfigurieren und benötigen dann Hilfe, oder schlimmer noch, können Sicherheitslücken öffnen, ohne dies zu merken.
- *Ausführen unsicherer Dienste* — Benutzer mit Root-Berechtigungen können unsichere Dienste, wie zum Beispiel FTP oder Telnet auf ihrem Rechner ausführen und dadurch Benutzernamen und Passwörter einem Risiko aussetzen, da diese im Klartext über das Netzwerk verschickt werden.
- *Als Root E-Mail-Anhänge öffnen* — Wenn auch selten, so gibt es doch E-Mail-Viren, die Linux angreifen. Dies wird jedoch nur dann zum Problem, wenn sie als Root ausgeführt werden.

2.1.4.2. Verwehren von Root-Zugriff

Falls ein Administrator aus diesen oder anderen Gründen den Benutzern keine Root-Rechte gewähren möchte, sollte das Root-Passwort geheim gehalten werden und der Zugriff auf Runlevel 1 oder Einzelbenutzermodus mithilfe eines Bootloader-Passworts verwehrt werden (siehe Abschnitt 2.1.2.2, „Bootloader-Passwörter“ für weitere Informationen diezbezüglich.)

Tabelle 2.1, „Methoden zum Deaktivieren des Root-Accounts“ zeigt Methoden, mit denen ein Administrator Anmeldungen als Root noch weiter verhindern kann:

Tabelle 2.1. Methoden zum Deaktivieren des Root-Accounts

Methode	Beschreibung	Effekt	Keine Auswirkung auf
Ändern der Root-Shell	Bearbeiten Sie die **`/etc/passwd`**-Datei und ändern Sie die Shell von **`/bin/bash`** auf **`/sbin/nologin`**.	Verhindert Zugang zur Root-Shell und protokolliert jegliche Versuche. Die folgenden Programme werden daran gehindert, auf den Root-Account zuzugreifen: · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · **`su`** · **`ssh`** · **`scp`** · **`sftp`**	Programme, die keine Shell benötigen, wie zum Beispiel FTP-Clients, Mail-Clients und viele setuid-Programme. Folgende Programme werden *nicht* daran gehindert, auf den Root-Account zuzugreifen: · **`sudo`** · FTP-Clients · E-Mail-Clients
Deaktivieren des Root-Zugriffs über Konsolengeräte (tty)	Eine leere **`/etc/securetty`**-Datei verhindert die Anmeldung als Root auf jeglichen am Computer angeschlossenen Geräten.	Verhindert den Zugriff auf den Root-Account über die Konsole oder das Netzwerk. Folgende Programme werden daran gehindert, auf den Root-Account zuzugreifen: · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · Andere Netzwerkdienste, die ein tty öffnen	Programme, die sich nicht als Root anmelden, jedoch administrative Aufgaben mittels setuid oder anderen Mechanismen ausführen. Folgende Programme werden *nicht* daran gehindert, auf den Root-Account zuzugreifen: · **`su`** · **`sudo`** · **`ssh`** · **`scp`** · **`sftp`**
Deaktivieren von SSH-Logins als Root	Bearbeiten Sie die Datei **`/etc/ssh/sshd_config`** und setzen Sie den **`PermitRootLogin`**-Parameter auf no	Verhindert den Root-Zugriff via OpenSSH-Suite-Tools. Folgende Programme werden daran gehindert, auf	Da dies nur die OpenSSH-Tool-Suite betrifft, sind keine anderen Programme von dieser Einstellung

Root	Parameter auf **no**.	den Root-Account zuzugreifen: · **ssh** · **scp** · **sftp**	betroffen.
Mit PAM den Root-Zugang zu Diensten einschränken.	Bearbeiten Sie die Datei für den Zieldienst im Verzeichnis **/etc/pam.d/**. Stellen Sie sicher, dass die **pam_listfile.so** zur Authentifizierung erforderlich ist. [a]	Verhindert den Root-Zugriff auf Netzwerkdienste, die PAM verwenden. Die folgenden Dienste werden daran gehindert, auf den Root-Account zuzugreifen: · FTP-Clients · E-Mail-Clients · **login** · **gdm** · **kdm** · **xdm** · **ssh** · **scp** · **sftp** · Jegliche Dienste, die PAM verwenden	Programme und Dienste, die PAM nicht berücksichtigen.

[a] Siehe Abschnitt 2.1.4.2.4, „Deaktivieren von PAM für Root“ für Einzelheiten.

2.1.4.2.1. Deaktivieren der Root-Shell

Um zu verhindern, dass sich Benutzer direkt als Root anmelden, kann der Systemadministrator die Shell des Root-Accounts in der **/etc/passwd**-Datei auf **/sbin/nologin** setzen. Dies verhindert Zugang zum Root-Account über Befehle, die eine Shell benötigen, wie zum Beispiel **su** oder **ssh**.

Wichtig

Programme, die keinen Zugang zur Shell benötigen, wie z. B. E-Mail-Clients oder der **sudo**-Befehl, können weiterhin auf den Root-Account zugreifen.

2.1.4.2.2. Deaktivieren von Root-Anmeldungen

Um den Zugang zum Root-Account noch weiter einzuschränken, können Administratoren Root-Anmeldungen an der Konsole verhindern, indem sie die Datei **/etc/securetty** bearbeiten. In dieser Datei werden alle Geräte aufgelistet, an denen sich der Root-Benutzer anmelden darf. Existiert die Datei nicht, darf sich der Root-Benutzer über jedes beliebige Kommunikationsgerät auf dem System anmelden, sei es über eine Konsole oder eine Raw-Netzwerkschnittstelle. Dies stellt ein Risiko dar, da ein Benutzer sich über Telnet am Computer als Root anmelden kann, wobei die Passwörter im Klartext über das Netzwerk versendet werden. Standardmäßig erlaubt die Red Hat Enterprise Linux Datei **/etc/securetty** dem Root-Benutzer nur, sich an der mit dem Rechner direkt verbundenen Konsole anzumelden. Um das Anmelden von Root zu verhindern, löschen Sie den Inhalt dieser Datei, indem Sie folgenden Befehl eingeben:

```
echo > /etc/securetty
```

Warnung

Eine leere **/etc/securetty**-Datei verhindert *nicht*, dass der Root-Benutzer sich von außen über die OpenSSH Toolsuite anmeldet, da die Konsole erst nach der Authentifizierung geöffnet wird.

2.1.4.2.3. Deaktivieren von Root SSH-Anmeldungen

Root-Anmeldungen über das SSH-Protokoll sind in Red Hat Enterprise Linux 6 standardmäßig deaktiviert. Falls diese Option jedoch zwischenzeitlich aktiviert wurde, kann sie wieder deaktiviert werden, indem Sie die Konfigurationsdatei des SSH-Daemons (**/etc/ssh/sshd_config**) bearbeiten. Ändern Sie folgende Zeile:

```
PermitRootLogin yes
```

zu:

```
PermitRootLogin no
```

Damit diese Änderungen wirksam werden, muss der SSH-Daemon neu gestartet werden. Führen Sie dazu den folgenden Befehl aus:

```
kill -HUP `cat /var/run/sshd.pid`
```

2.1.4.2.4. Deaktivieren von PAM für Root

PAM ermöglicht durch das **/lib/security/pam_listfile.so**-Modul eine größere Flexibilität in der Ablehnung bestimmter Accounts. Mithilfe dieses Moduls kann der Administrator eine Liste von Benutzern festlegen, denen die Anmeldung nicht gestattet ist. Unten finden Sie ein Beispiel, wie das Modul für den **vsftpd**-FTP-Server in der **/etc/pam.d/vsftpd** PAM-Konfigurationsdatei verwendet werden kann (das **** Zeichen am Ende der ersten Zeile im folgenden Beispiel ist *nicht* nötig, wenn die Direktive auf einer Zeile steht):

```
auth required /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Dies weist PAM an, die Datei **/etc/vsftpd.ftpusers** zu lesen und allen hier aufgeführten Benutzern

Zugang zum Dienst zu verbieten. Der Administrator kann den Namen dieser Datei ändern und separate Listen für jeden Dienst oder eine einzige zentrale Liste für die Zugriffsverweigerung für mehrere Dienste führen.

Wenn der Administrator den Zugang zu mehreren Diensten verbieten will, kann eine ähnliche Zeile zu den PAM-Konfigurationsdateien wie z. B. **/etc/pam.d/pop** und **/etc/pam.d/imap** für Mail-Clients oder **/etc/pam.d/ssh** für SSH-Clients hinzugefügt werden.

Weitere Informationen über PAM finden Sie unter *Managing Single Sign-On and Smart Cards*.

2.1.4.3. Beschränken des Root-Zugangs

Statt dem Root-Benutzer den Zugriff völlig zu verwehren, kann der Administrator den Zugriff ausschließlich über setuid-Programme wie **su** oder **sudo** gewähren.

2.1.4.3.1. Der su-Befehl

Wenn ein Benutzer den **su**-Befehl ausführt, wird er nach dem Root-Passwort gefragt und erhält nach erfolgreicher Authentifizierung ein Root-Shell-Prompt.

Nach der Anmeldung über den **su**-Befehl *ist* der Benutzer tatsächlich der Root-Benutzer und hat vollständigen administrativen Zugriff auf das System [13]. Nachdem der Benutzer auf diese Weise zum Root-Benutzer geworden ist, kann er mit dem Befehl **su** zu jedem anderen Benutzer im System wechseln, ohne nach einem Passwort gefragt zu werden.

Da dieses Programm sehr mächtig ist, sollten Administratoren im Unternehmen den Zugang zu diesem Befehl beschränken.

Einer der einfachsten Wege dazu ist es, Benutzer zur administrativen Gruppe mit dem Namen *wheel* hinzuzufügen. Hierzu geben Sie den folgenden Befehl als Root ein:

```
usermod -G wheel <username>
```

Ersetzen Sie in diesem Befehl ***<username>*** durch den Benutzernamen, den Sie zur **wheel**-Gruppe hinzufügen möchten.

Sie können auch die grafische **Benutzerverwaltung** nutzen, um wie nachfolgend beschrieben Gruppenmitgliedschaften zu ändern. Beachten Sie, dass Sie zur Durchführung dieses Verfahrens über Administratorrechte verfügen müssen.

1. Klicken Sie im **System**-Menü auf der oberen Menüleiste auf **Administration** und anschließend auf **Benutzer und Gruppen**, um die Benutzerverwaltung anzuzeigen. Alternativ können Sie dazu auch den Befehl **system-config-users** an einem Shell-Prompt eingeben.
2. Klicken Sie auf den **Benutzer**-Reiter und wählen Sie den gewünschten Benutzer aus der Liste aus.
3. Klicken Sie auf **Eigenschaften** in der Werkzeugleiste, um das Dialogfeld mit den Benutzereigenschaften anzuzeigen (oder wählen Sie **Eigenschaften** aus dem **Datei**-Menü).
4. Klicken Sie auf den **Gruppen**-Reiter, markieren Sie das Auswahlkästchen für die "wheel"-Gruppe und klicken Sie anschließend auf **OK**. Siehe Abbildung 2.2, „Hinzufügen von Benutzern zur "wheel"-Gruppe.“.

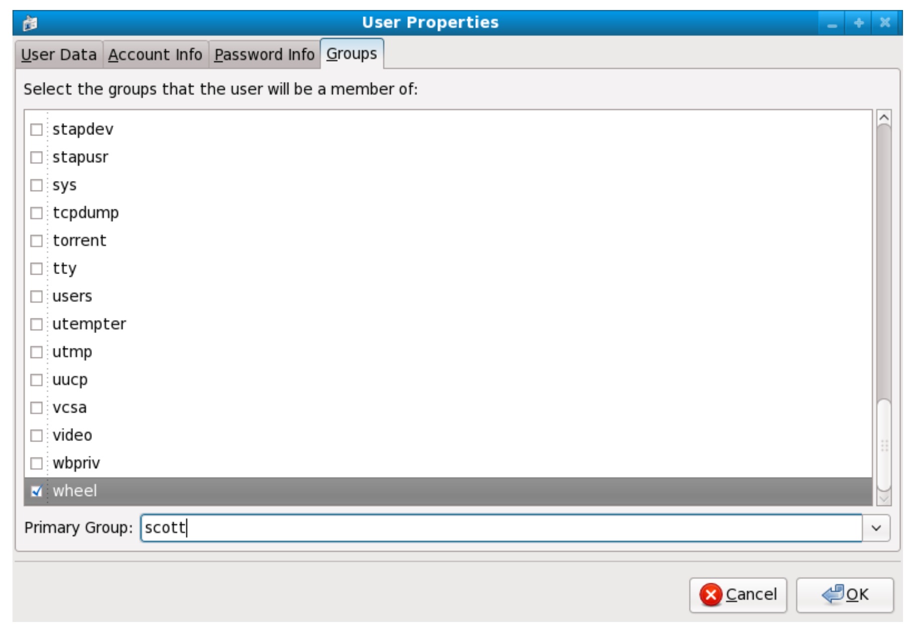

Abbildung 2.2. Hinzufügen von Benutzern zur "wheel"-Gruppe.

Öffnen Sie die PAM-Konfigurationsdatei für **su** (**/etc/pam.d/su**) in einem Texteditor und entfernen Sie die Kommentierung # von der folgenden Zeile:

```
auth  required /lib/security/$ISA/pam_wheel.so use_uid
```

Hierdurch können nur Mitglieder der administrativen Gruppe **wheel** dieses Programm nutzen.

Anmerkung

Der Root-Benutzer ist standardmäßig Mitglied der **wheel**-Gruppe.

2.1.4.3.2. Der sudo-Befehl

Der **sudo**-Befehl bietet eine weitere Methode, Benutzern administrativen Zugang zu gewähren. Wenn ein vertrauenswürdiger Benutzer einem administrativen Befehl den **sudo**-Befehl voranstellt, wird dieser nach *seinem eigenen* Passwort gefragt. Nach erfolgreicher Authentifizierung und vorausgesetzt, dass der Befehl erlaubt ist, wird der administrative Befehl wie von einem Root-Benutzer ausgeführt.

Das grundlegende Format des **sudo**-Befehls lautet wie folgt:

```
sudo <command>
```

Im obigen Beispiel würde ***<command>*** durch einen Befehl ersetzt, der normalerweise für den Root-Benutzer reserviert ist, wie z. B. **mount**.

Wichtig

Benutzer des **sudo**-Befehls sollten sicherstellen, dass sie sich abmelden, bevor Sie sich von Ihrem Rechner entfernen, da Sudoers den Befehl innerhalb von fünf Minuten erneut ausführen können, ohne nach einem Passwort gefragt zu werden. Diese Einstellung kann mithilfe der Konfigurationsdatei **/etc/sudoers** geändert werden.

Der **sudo**-Befehl ermöglicht einen hohen Grad an Flexibilität. So können z. B. nur Benutzer, die in der Konfigurationsdatei **/etc/sudoers** aufgeführt sind, den Befehl **sudo** ausführen; dieser Befehl wird dann in der Shell *des Benutzers* ausgeführt, und nicht in der Root-Shell. Dies bedeutet, das die Root-Shell vollständig deaktiviert werden kann, wie in Abschnitt 2.1.4.2.1, „Deaktivieren der Root-Shell “ gezeigt.

Der **sudo**-Befehl liefert auch ein umfangreiches Audit-Protokoll. Jede erfolgreiche Authentifizierung wird in die Datei **/var/log/messages** geschrieben, und der ausgeführte Befehl samt Benutzername wird in die Datei **/var/log/secure** geschrieben.

Ein weiterer Vorteil des **sudo**-Befehls ist, dass ein Administrator verschiedenen Benutzern Zugang zu bestimmten Befehlen basierend auf deren Bedürfnissen geben kann.

Administratoren, die die **sudo**-Konfigurationsdatei **/etc/sudoers** bearbeiten wollen, sollten dazu den Befehl **visudo** verwenden.

Um einem Benutzer umfassende administrative Rechte zu geben, geben Sie **visudo** ein und fügen Sie eine Zeile ähnlich der folgenden in den Abschnitt für die Benutzerrechte ein:

```
juan ALL=(ALL) ALL
```

Dieses Beispiel besagt, dass der Benutzer **juan** den **sudo**-Befehl auf jedem Host für jeden Befehl ausführen kann.

Das nachfolgende Beispiel veranschaulicht die möglichen Feinheiten bei der Konfiguration von **sudo**:

```
%users localhost=/sbin/shutdown -h now
```

Dieses Beispiel besagt, dass jeder Benutzer den Befehl **/sbin/shutdown -h now** ausführen kann, solange dieser auf der Konsole ausgeführt wird.

Die **sudoers**-Handbuchseite enthält eine detaillierte Liste aller Optionen für diese Datei.

2.1.5. Verfügbare Netzwerkdienste

Während der Benutzerzugriff auf administrative Kontrollen hauptsächlich für Systemadministratoren innerhalb eines Unternehmens ein wichtiges Thema ist, ist die Kontrolle der Netzwerkdienste dagegen für jeden von höchster Priorität, der ein Linux-System verwaltet und verwendet.

Viele Dienste unter Red Hat Enterprise Linux verhalten sich als Netzwerkserver. Wenn ein Netzwerkdienst auf einem Rechner ausgeführt wird, horcht eine Server-Applikation (auch *Daemon* genannt) auf einem oder mehreren Ports auf Verbindungen. Jeder dieser Server sollte als potenzielle Angriffsstelle betrachtet werden.

2.1.5.1. Risiken für Dienste

Netzwerkdienste können viele Risiken für Linuxsysteme mit sich bringen. Nachfolgend finden Sie eine Liste der Hauptprobleme:

- *Denial-of-Service-Angriff (DoS)* — Indem ein System mit Anfragen überflutet wird, kann ein Denial-of-Service-Angriff ein System zum völligen Stillstand bringen, da das System versucht, jede Anfrage zu protokollieren und zu beantworten.
- *Distributed-Denial-of-Service-Angriff (DDoS)* — Eine Art von DoS-Angriff, bei dem mehrere infizierte Rechner (oft Tausende) missbraucht werden, um einen koordinierten Angriff auf einen Dienst durchzuführen und diesen mit Anfragen zu überfluten.
- *Skript-Angriff* — Wenn ein Server Skripte zum Ausführen von serverseitigen Aufgaben verwendet, wie es Webserver gewöhnlich tun, kann ein Cracker durch nicht-sachgemäß erstellte Skripte einen Angriff initiieren. Diese Skript-Angriffe können zu einem Pufferüberlauf führen oder es dem Angreifer ermöglichen, Dateien auf dem Server zu ändern.
- *Pufferüberlauf-Angriff* — Dienste, die sich auf Ports 0 bis 1023 verbinden, müssen als administrativer Benutzer ausgeführt werden. Hat die Applikation einen Pufferüberlauf, kann ein Angreifer Zugang zum System erlangen als der Benutzer, der den Daemon ausführt. Da Pufferüberläufe existieren, können Cracker mit automatisierten Tools das System auf Schwachstellen prüfen. Sobald diese dann Zugang zum System haben, können sie mithilfe automatisierter Root-Kits den Zugang zum System aufrecht erhalten.

Anmerkung

Die Bedrohung durch Schwachstellen, die bei Pufferüberläufen entstehen, wird in Red Hat Enterprise Linux durch *ExecShield* entschärft, einer ausführbaren Speichersegmentation und Schutztechnologie, unterstützt von x86-kompatiblen Einzelprozessor- und Multiprozessor-Kernels. ExecShield reduziert das Risiko eines Pufferüberlaufs, indem virtueller Speicher in ausführbare und nicht-ausführbare Segmente unterteilt wird. Jeglicher Programmcode, der versucht, sich außerhalb des ausführbaren Segments auszuführen (wie z. B. schädlicher Code, der unter Ausnutzung einer Pufferüberlauf-Sicherheitslücke eingeschleust wurde), löst einen Segmentierungsfehler aus und wird beendet.
Execshield beinhaltet auch Unterstützung für *No eXecute* (NX) Technologie auf AMD64-Plattformen und *eXecute Disable* (XD) Technologie auf Itanium und Intel® 64-Systemen. Diese Technologien arbeiten zusammen mit ExecShield, um schädlichen Code davon abzuhalten, im ausführbaren Bereich des virtuellen Speichers mit einer Granularität von 4 KB ausführbaren Codes abzulaufen, wodurch das Risiko eines heimlichen Angriffs unter Ausnutzung eines Pufferüberlaufs verringert wird.

Wichtig

Um die Angriffsfläche des Netzwerks zu verringern, sollten alle nicht genutzten Dienste ausgeschaltet werden.

2.1.5.2. Identifizieren und Konfigurieren von Diensten

Zur Erhöhung der Sicherheit sind die meisten Netzwerkdienste, die mit Red Hat Enterprise Linux installiert werden, standardmäßig deaktiviert. Es gibt jedoch einige nennenswerte Ausnahmen:

- **`cupsd`** — Der standardmäßige Druckerserver für Red Hat Enterprise Linux.
- **`lpd`** — Ein alternativer Druckerserver.
- **`xinetd`** — Ein Super-Server, der die Verbindungen zu einer Reihe untergeordneter Server, wie zum

Beispiel **gssftp** und **telnet** steuert.

- **sendmail** — Der Sendmail *Mail Transport Agent* (MTA) ist standardmäßig aktiviert, horcht jedoch nur auf Verbindungen von localhost.
- **sshd** — Der OpenSSH Server, ein sicherer Ersatz für Telnet.

Bei der Entscheidung, ob diese Dienste aktiviert bleiben sollen, sollten Sie mit gesundem Menschenverstand handeln und Vorsicht walten lassen. Wenn Sie zum Beispiel keinen Drucker anschließen, sollten Sie **cupsd** nicht ausführen. Das gleiche gilt für **portmap**. Wenn Sie keine NFSv3-Datenträger einhängen oder NIS (den **ypbind**-Dienst) nicht verwenden, sollte Portmap deaktiviert werden.

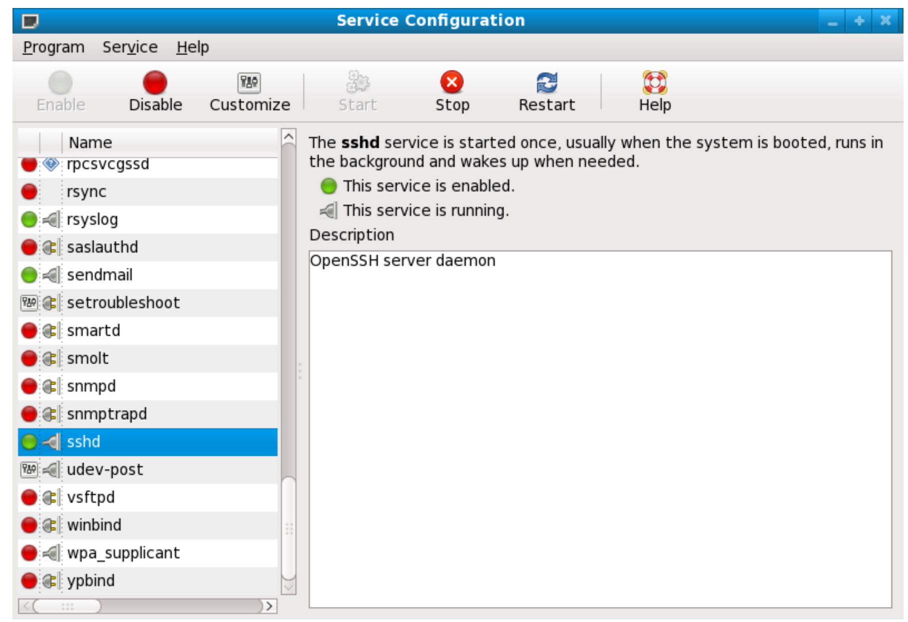

Abbildung 2.3. Tool zur Dienstkonfiguration

Wenn Sie sich nicht sicher sind, welchen Zweck ein Dienst hat, finden Sie im **Tool zur Dienstkonfiguration** ein Beschreibungsfeld, wie in Abbildung 2.3, „Tool zur Dienstkonfiguration“ dargestellt, das zusätzliche Informationen liefert.

Das reine Überprüfen, welche Netzwerkdienste zum Bootzeitpunkt verfügbar sind, ist jedoch nicht genug. Sie sollten auch prüfen, welche Ports offen sind und horchen. Weitere Informationen zu diesem Thema finden Sie unter Abschnitt 2.2.8, „Überprüfen der horchenden Ports“.

2.1.5.3. Unsichere Dienste

Jeder Netzwerkdienst ist potenziell unsicher. Aus diesem Grund ist es wichtig, nicht benötigte Dienste zu deaktivieren. Zudem werden regelmäßig neue Schwachstellen in Diensten gefunden und gepatcht, weshalb es ebenfalls enorm wichtig ist, die Pakete für Netzwerkdienste immer auf dem neuesten Stand zu halten. Weitere Informationen hierzu finden Sie unter Abschnitt 1.5, „Sicherheitsaktualisierungen“.

Einige Netzwerkprotokolle sind von Natur aus unsicherer als andere. Dazu gehören insbesondere Dienste mit folgenden Merkmalen:

- *Unverschlüsselte Übertragung von Benutzernamen und Passwörtern über ein Netzwerk* — Viele ältere Protokolle, wie beispielsweise Telnet und FTP, verschlüsseln die Authentifizierung nicht und sollten möglichst deaktiviert werden.
- *Unverschlüsselte Übertragung von sensiblen Daten über ein Netzwerk* — Viele Protokolle übertragen Daten unverschlüsselt über das Netzwerk. Zu diesen Protokollen gehört unter anderem Telnet, FTP,

HTTP und SMTP. Viele Netzwerkdateisysteme wie z. B. NFS und SMB übertragen ebenfalls Informationen unverschlüsselt über das Netzwerk. Es liegt in der Verantwortung des Benutzers, einzuschränken, welche Art von Daten bei der Verwendung dieser Protokolle übertragen werden.

Auch Remote-Speicherabbildungsdienste wie **netdump** übertragen den Speicherinhalt unverschlüsselt über das Netzwerk. Speicherauszüge können Passwörter, oder schlimmer noch, Datenbankeinträge und andere sensible Informationen enthalten.

Andere Dienste wie **finger** und **rwhod** geben Informationen über Benutzer im System preis.

Zu den von Natur aus unsicheren Diensten gehören unter anderem **rlogin**, **rsh**, **telnet** und **vsftpd**.

Alle Remote-Login- und Shell-Programme (**rlogin**, **rsh** und **telnet**) sollten zugunsten von SSH vermieden werden. Siehe Abschnitt 2.1.7, „Kommunikationstools mit verbesserter Sicherheit“ für weitere Informationen über **sshd**.

FTP ist von Natur aus nicht ganz so gefährlich für die Sicherheit des Systems wie Remote-Shells, FTP-Server müssen jedoch sorgfältig konfiguriert und überwacht werden, um Probleme zu vermeiden. Weitere Informationen über das Sichern von FTP-Servern finden Sie unter Abschnitt 2.2.6, „Sichern von FTP“.

Dienste, die sorgfältig implementiert und hinter einer Firewall platziert werden sollten, umfassen:

- **finger**
- **authd** (in früheren Red Hat Enterprise Linux Releases **identd** genannt)
- **netdump**
- **netdump-server**
- **nfs**
- **rwhod**
- **sendmail**
- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

Weitere Informationen zur Sicherung von Netzwerkdiensten finden Sie unter Abschnitt 2.2, „Server-Sicherheit“.

Im nächsten Abschnitt werden Tools für das Einrichten einer Firewall beschrieben.

2.1.6. Persönliche Firewalls

Sobald die *notwendigen* Netzwerkdienste konfiguriert sind, ist es wichtig, eine Firewall zu implementieren.

Wichtig

Sie sollten die notwendigen Netzwerkdienste konfigurieren und eine Firewall implementieren, *bevor* Sie sich mit dem Internet oder anderen nicht vertrauenswürdigen Netzwerken verbinden.

Firewalls verhindern, dass Netzwerkpakete Zugriff auf die Netzwerkschnittstelle des Systems erhalten.

Wird eine Anfrage an einen Port gestellt, der von einer Firewall geschützt ist, wird diese Anfrage ignoriert. Horcht ein Dienst auf einen dieser blockierten Ports, kann dieser Dienst die Pakete nicht empfangen, und ist somit effektiv deaktiviert. Aus diesem Grund sollte man bei der Konfiguration einer Firewall darauf achten, dass der Zugang zu nicht benutzten Ports blockiert wird, Ports für konfigurierte Dienste jedoch offen bleiben.

Für die meisten Benutzer ist das beste Tool zur Konfiguration einer einfachen Firewall das einfache, grafische Firewall-Konfigurationstool, das standardmäßig in Red Hat Enterprise Linux enthalten ist: **Tool zur Firewall-Konfiguration** (**system-config-securitylevel**). Dieses Tool erzeugt breite **iptables**-Regeln für eine allgemeine Firewall, unter Verwendung einer grafischen Benutzeroberfläche.

Weitere Informationen zur Verwendung dieser Applikation und ihrer verfügbaren Optionen finden Sie unter Abschnitt 2.5.2, „Grundlegende Firewall-Konfiguration“

Für fortgeschrittene Benutzer und Server-Administratoren ist wahrscheinlich die manuelle Konfiguration einer Firewall mittels **iptables** die bessere Wahl. Weitere Informationen finden Sie unter Abschnitt 2.5, „Firewalls“. Einen umfassenden Leitfaden zum **iptables**-Befehl finden Sie unter Abschnitt 2.6, „IPTables“.

2.1.7. Kommunikationstools mit verbesserter Sicherheit

Mit wachsender Verbreitung und Beliebtheit des Internets wächst auch das Risiko, dass Kommunikationsdatenverkehr abgefangen wird. In den letzten Jahren wurden Tools entwickelt, die jegliche Kommunikation bei der Übertragung über das Netzwerk verschlüsseln.

Red Hat Enterprise Linux 6 wird mit zwei einfachen Tools geliefert, die Verschlüsselungsalgorithmen auf höchster Ebene (basierend auf öffentlichen Schlüsseln) zum Schutz der Daten bei der Übertragung im Netzwerk verwenden.

- *OpenSSH* — Eine offene Implementierung des SSH-Protokolls zur Verschlüsselung von Netzwerkkommunikation.
- *Gnu Privacy Guard (GPG)* — Eine offene Implementierung der PGP (Pretty Good Privacy) Verschlüsselungsapplikation zur Verschlüsselung von Daten.

OpenSSH ist eine sichere Methode für den Zugang zur einer entfernten Maschine und ersetzt ältere, unverschlüsselte Dienste wie **telnet** und **rsh**. OpenSSH umfasst einen Netzwerkdienst namens **sshd** und drei Befehlszeilen-Client-Applikationen:

- **ssh** — Ein sicherer Client für den Zugriff auf Remote-Konsolen.
- **scp** — Ein sicherer Befehl für Remote-Copy.
- **sftp** — Ein sicherer Pseudo-FTP-Client, der interaktive Dateiübertragung ermöglicht.

Siehe Abschnitt 3.6, „Secure Shell“ für weitere Informationen über OpenSSH.

Wichtig

Obwohl der **sshd**-Dienst von Natur aus sicher ist, *muss* dieser Dienst auf dem neuesten Stand gehalten werden, um Sicherheitsgefährdungen zu vermeiden. Unter Abschnitt 1.5, „Sicherheitsaktualisierungen“ finden Sie weitere Informationen zu diesem Thema.

GPG ist eine mögliche Methode, um die private E-Mail-Kommunikation sicherzustellen. Es kann zum Versenden sensibler Daten per E-Mail über öffentliche Netzwerke sowie zum Schutz sensibler Daten auf Festplatten eingesetzt werden.

2.2. Server-Sicherheit

Wenn ein System als Server in einem öffentlichen Netzwerk verwendet wird, stellt es ein Ziel für Angreifer dar. Aus diesem Grund ist das Abhärten des Systems und Sperren von Diensten von erheblicher Bedeutung für den Systemadministrator.

Bevor Sie die Details eines bestimmten Themas erforschen, sehen Sie sich die folgenden, allgemeinen Hinweise für das Erhöhen der Server-Sicherheit an:

- Halten Sie alle Dienste auf dem neuesten Stand, um vor den neuesten Bedrohungen geschützt zu sein.
- Verwenden Sie nach Möglichkeit sichere Protokolle.
- Wenn möglich, sollte immer nur eine Maschine eine Art von Netzwerkdienst bereitstellen.
- Überwachen Sie alle Server sorgfältig auf verdächtige Aktivitäten.

2.2.1. Sichern von Diensten mit TCP-Wrappern und xinetd

TCP-Wrapper bieten Zugriffskontrolle für eine Reihe von Diensten. Die meisten modernen Netzwerkdienste, wie z. B. SSH, Telnet und FTP, verwenden TCP-Wrapper, die als Wachposten zwischen einer eingehenden Anfrage und dem angefragten Dienst stehen.

Die Vorteile von TCP-Wrappern werden noch erweitert, wenn diese zusammen mit **xinetd** verwendet werden, einem Super-Serverdienst, der zusätzliche Zugriffs-, Protokollierungs-, Binding-, Umleitungs- und Ressourcenkontrolle bietet.

Anmerkung

Es ist von Vorteil, die IPTables-Firewall-Regeln zusammen mit TCP-Wrappern und **xinetd** zu verwenden, um eine Redundanz innerhalb der Dienst-Zugangskontrollen zu erreichen. Für mehr Information über das Einrichten von Firewalls mit IPTables-Befehlen siehe Abschnitt 2.5, „Firewalls“.

Die folgenden Abschnitte setzen ein grundlegendes Wissen über das jeweilige Thema voraus und konzentrieren sich daher auf spezielle Sicherheitsoptionen.

2.2.1.1. Erhöhung der Sicherheit mit TCP-Wrappern

TCP-Wrapper können viel mehr als nur Zugriffe auf Dienste verweigern. In diesem Abschnitt wird erläutert, wie TCP-Wrapper zum Versenden von Verbindungs-Bannern, Warnen vor Angriffen von bestimmten Hosts und Erweitern der Protokollierungsfunktionalität eingesetzt werden können. Mehr Informationen über die Funktionalität und Kontrollsprache der TCP-Wrapper finden Sie auf der **hosts_options**-Handbuchseite. Werfen Sie zudem einen Blick auf die **xinetd.conf**-Handbuchseite, erhältlich online unter http://linux.die.net/man/5/xinetd.conf, für Informationen über verfügbare Flags, die Sie als Optionen auf einen Dienst anwenden können.

2.2.1.1.1. TCP-Wrapper und Verbindungsbanner

Benutzern beim Verbinden mit einem Dienst ein einschüchterndes Banner anzuzeigen, ist eine gute Methode, um potenzielle Angreifer wissen zu lassen, dass sie es mit einem aufmerksamen Systemadministrator zu tun haben. Zugleich können Sie auf diese Weise steuern, welche Informationen über das System den Benutzern gezeigt werden. Um ein TCP-Wrapper-Banner für einen Dienst zu implementieren, verwenden Sie die Option **banner**.

In diesem Beispiel wird ein Banner für **vsftpd** implementiert. Erstellen Sie zunächst einmal eine Bannerdatei. Es ist unerheblich, wo diese sich auf dem System befindet, muss aber den gleichen Namen wie der Daemon tragen. In diesem Beispiel heißt die Datei **/etc/banners/vsftpd** und enthält die folgende Zeile:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being removed.
```

Der **%c**-Token liefert eine Reihe von Client-Informationen wie den Benutzernamen und Hostnamen, oder den Benutzernamen und die IP-Adresse, um die Verbindung noch abschreckender zu machen.

Damit dieses Banner bei eingehenden Verbindungen angezeigt wird, fügen Sie die folgende Zeile in die Datei **/etc/hosts.allow** ein:

```
vsftpd : ALL : banners /etc/banners/
```

2.2.1.1.2. TCP-Wrapper und Warnung vor Angriffen

Wenn ein bestimmter Host oder ein Netzwerk bei einem Angriff auf den Server erwischt wurde, können TCP-Wrapper mit der **spawn**-Direktive vor weiteren Angriffen von diesem Host oder Netzwerk warnen.

In diesem Beispiel gehen wir davon aus, dass ein Cracker vom 206.182.68.0/24 Netzwerk bei einem Angriffsversuch auf den Server erwischt wurde. Indem Sie die folgende Zeile in die Datei **/etc/hosts.deny** einfügen, wird der Verbindungsversuch abgewiesen und in einer speziellen Datei aufgezeichnet:

```
ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder_alert
```

Der **%d**-Token gibt den Namen des Dienstes an, auf den der Angreifer zugreifen wollte.

Um die Verbindung zu erlauben und diese aufzuzeichnen, fügen Sie die **spawn**-Direktive in die Datei **/etc/hosts.allow** ein.

Anmerkung

Da die **spawn**-Direktive jeden beliebigen Shell-Befehl ausführt, können Sie ein spezielles Skript schreiben, das den Administrator im Falle eines Verbindungsversuchs eines bestimmten Clients mit dem Server benachrichtigt oder eine Reihe von Befehlen ausführt.

2.2.1.1.3. TCP-Wrapper und erweiterte Protokollierung

Falls bestimmte Verbindungstypen Anlass zu größerer Sorge geben als andere, kann die Protokollierungsebene für den jeweiligen Dienst über die Option **severity** angehoben werden.

Lassen Sie uns für dieses Beispiel annehmen, dass jeder, der eine Verbindung zu Port 23 (dem Telnet-Port) auf einem FTP-Server herstellen will, ein Cracker ist. Um dies zu kennzeichnen, fügen Sie ein **emerg**-Flag anstelle des Standard-Flags **info** in die Protokolldatei ein und verweigern Sie die Verbindung.

Fügen Sie dazu die folgende Zeile in die Datei **/etc/hosts.deny** ein:

```
in.telnetd : ALL : severity emerg
```

Dadurch wird die standardmäßige **authpriv**-Protokollierungs-Facility verwendet, jedoch wird die Priorität vom Standardwert **info** auf **emerg** angehoben, wodurch Protokollnachrichten direkt auf der Konsole ausgegeben werden.

2.2.1.2. Erhöhen der Sicherheit mit xinetd

In diesem Abschnitt wird erläutert, wie **xinetd** dazu eingesetzt werden kann, einen so genannten Trap-Dienst einzurichten sowie die verfügbaren Ressourcen für jeden **xinetd**-Dienst zu kontrollieren. Das Setzen von Ressourcengrenzen kann dabei helfen, *Denial of Service* (DoS)-Angriffe zu unterbinden. Eine Liste der verfügbaren Optionen finden Sie auf den Handbuchseiten zu **xinetd** und **xinetd.conf**.

2.2.1.2.1. Aufstellen einer Falle

Ein wichtiges Feature von **xinetd** ist die Fähigkeit, Hosts zu einer globalen **no_access**-Liste hinzufügen zu können. Den Hosts auf dieser Liste werden Verbindungen zu Diensten, die von **xinetd** verwaltet werden, für einen bestimmten Zeitraum oder bis **xinetd** neu gestartet wird, verweigert. Dies wird durch den **SENSOR**-Parameter erreicht. Mithilfe dieses einfachen Verfahrens können Sie Hosts blockieren, die den Server auf offene Ports absuchen.

Der erste Schritt für das Einrichten von **SENSOR** ist die Auswahl eines Dienstes, den Sie voraussichtlich nicht anderweitig brauchen werden. In diesem Beispiel wird Telnet ausgewählt.

Bearbeiten Sie die Datei **/etc/xinetd.d/telnet** und ändern Sie die Zeile **flags** folgendermaßen um:

```
flags           = SENSOR
```

Fügen Sie folgende Zeile hinzu:

```
deny_time       = 30
```

Dadurch werden einem Host alle weitere Verbindungsversuche auf diesem Port für 30 Minuten verweigert. Andere gültige Werte für das **deny_time**-Attribut sind FOREVER, wodurch eine Verbindung solange verweigert wird, bis **xinetd** neu gestartet wird, und NEVER, wodurch die Verbindung zugelassen und protokolliert wird.

Die letzte Zeile sollte Folgendes enthalten:

```
disable         = no
```

Dadurch wird die Falle selbst aktiviert.

Obwohl **SENSOR** eine gute Methode ist, Verbindungen von böswilligen Hosts zu erkennen und zu stoppen, hat es jedoch zwei Nachteile:

- Es hilft nicht gegen heimliches Scannen (Stealth Scans).
- Ein Angreifer, der weiß, dass ein **SENSOR** aktiviert ist, kann eine DoS-Attacke gegen bestimmte Hosts ausführen, indem er ihre IP-Adressen fälscht und sich mit dem verbotenen Port verbindet.

2.2.1.2.2. Kontrollieren von Server-Ressourcen

Ein weiteres, wichtiges Feature von **xinetd** ist die Fähigkeit, für die von ihm kontrollierten Dienste Ressourcengrenzen festzulegen.

Dies wird durch die folgenden Direktiven erreicht:

- **cps = <number_of_connections> <wait_period>** — Begrenzt die Frequenz der eingehenden Verbindungen. Diese Direktive akzeptiert zwei Parameter:
  - **<number_of_connections>** — Die Anzahl der zu verarbeitenden Verbindungen pro Sekunde. Falls die Frequenz der eingehenden Verbindungen diesen Wert überschreitet, wird der Dienst zeitweise deaktiviert. Der Standardwert ist fünfzig (50).
  - **<wait_period>** — Gibt die Anzahl der Sekunden an, die gewartet werden soll, bevor der Dienst nach dessen Deaktivierung neu gestartet werden soll. Die Standardzeitspanne beträgt zehn (10) Sekunden.
- **instances = <number_of_connections>** — Gibt die Gesamtzahl aller erlaubten Verbindungen zu einem Dienst an. Diese Direktive akzeptiert entweder einen ganzzahligen Wert oder **UNLIMITED**.
- **per_source = <number_of_connections>** — Gibt die Anzahl der Verbindungen an, die pro Host zu einem Dienst erlaubt sind. Diese Direktive akzeptiert entweder einen ganzzahligen Wert oder **UNLIMITED**.
- **rlimit_as = <number[K|M]>** — Gibt die Größe des Speicheradressraums in Kilobyte oder Megabyte an, die der Dienst in Anspruch nehmen kann kann. Diese Direktive akzeptiert entweder einen ganzzahligen Wert oder **UNLIMITED**.
- **rlimit_cpu = <number_of_seconds>** — Gibt die Zeit in Sekunden an, die ein Dienst die CPU beanspruchen kann. Diese Direktive akzeptiert entweder einen ganzzahligen Wert oder **UNLIMITED**.

Mithilfe dieser Direktiven kann verhindert werden, dass ein einziger **xinetd**-Dienst das gesamte System überschwemmt und einen Denial-of-Service verursacht.

2.2.2. Sichern von Portmap

Der **portmap**-Dienst ist ein Daemon zur dynamischen Port-Zuweisung für RPC-Dienste wie NIS und NFS. Er besitzt schwache Authentifizierungsmechanismen und hat die Fähigkeit, einen großen Bereich an Ports für die von ihm kontrollierten Dienste zuzuweisen. Aus diesen Gründen ist Portmap schwer zu sichern.

Anmerkung

Das Sichern von **portmap** betrifft lediglich NFSv2- und NFSv3-Implementationen, da Portmap für NFSv4 nicht mehr länger erforderlich ist. Wenn Sie einen NFSv2- oder NFSv3-Server implementieren möchten, dann ist **portmap** demnach erforderlich und der folgende Abschnitt für Sie wichtig.

Falls Sie RPC-Dienste ausführen, sollten Sie die folgenden Grundregeln beachten.

2.2.2.1. Schützen von Portmap mit TCP-Wrappern

Es ist wichtig, TCP-Wrapper zur Einschränkung des Zugriffs von Netzwerken und Hosts auf den **portmap**-Dienst einzusetzen, da Portmap selbst keine integrierte Authentifizierungsmöglichkeit bietet.

Des Weiteren sollten Sie *nur* IP-Adressen verwenden, um den Zugriff auf den Dienst einzuschränken. Vermeiden Sie den Gebrauch von Hostnamen, da sie durch DNS-Poisoning und andere Methoden gefälscht werden können.

2.2.2.2. Schützen von Portmap mit IPTables

Um den Zugriff auf den **portmap**-Dienst weiter einzuschränken, ist es sinnvoll, IPTables-Regeln zum Server hinzuzufügen, die den Zugriff auf bestimmte Netzwerke einschränken.

Nachfolgend finden Sie zwei Beispiele für IPTables-Befehle. Der erste Befehl erlaubt TCP-Verbindungen auf Port 111 (welcher vom **portmap**-Dienst verwendet wird) vom 192.168.0/24 Netzwerk. Der zweite Befehl erlaubt TCP-Verbindungen auf demselben Port vom lokalen Host, was für den **sgi_fam**-Dienst für **Nautilus** benötigt wird. Alle anderen Pakete werden abgelehnt.

```
iptables -A INPUT -p tcp ! -s 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s 127.0.0.1  --dport 111 -j ACCEPT
```

Um auf gleiche Weise UDP-Datenverkehr einzuschränken, verwenden Sie den folgenden Befehl.

```
iptables -A INPUT -p udp ! -s 192.168.0.0/24  --dport 111 -j DROP
```

Anmerkung

Siehe auch Abschnitt 2.5, „Firewalls“ für weitere Informationen zum Einrichten von Firewalls mit dem IPTables-Befehl.

2.2.3. Sichern von NIS

Der *Network Information Service* (NIS) ist ein RPC-Dienst namens **ypserv**, der zusammen mit **portmap** und anderen zugehörigen Diensten verwendet wird, um Informationen zu Benutzernamen, Passwörtern und anderen sensiblen Daten an jeden beliebigen Computer innerhalb dessen Domain weiterzugeben.

Ein NIS-Server besteht aus mehreren Applikationen, unter anderem:

- **/usr/sbin/rpc.yppasswdd** — Auch **yppasswdd**-Dienst genannt. Dieser Daemon ermöglicht es Benutzern, ihre NIS-Passwörter zu ändern.
- **/usr/sbin/rpc.ypxfrd** — Auch **ypxfrd**-Dienst genannt. Dieser Daemon ist für den NIS-Map-Transfer über das Netzwerk verantwortlich.
- **/usr/sbin/yppush** — Diese Applikation verbreitet geänderte NIS-Datenbanken an mehrere NIS-Server.
- **/usr/sbin/ypserv** — Dies ist der NIS-Server-Daemon.

An heutigen Standards gemessen ist NIS als eher unsicher einzustufen. Es besitzt keine Host-Authentifizierungsmechanismen und überträgt Informationen, einschließlich Passwort-Hashes, unverschlüsselt über das Netzwerk. Aus diesem Grund müssen Sie beim Einrichten eines Netzwerks mit NIS äußerste Vorsicht walten lassen. Dadurch, dass die Standardkonfiguration von NIS von Natur aus unsicher ist, wird die Angelegenheit noch weiter verkompliziert.

Es wird empfohlen, dass Sie vor der Implementierung eines NIS-Servers zuerst den **portmap**-Dienst wie in Abschnitt 2.2.2, „Sichern von Portmap“ beschrieben sichern und dann weitere Bereiche wie z. B. Netzwerkplanung angehen.

2.2.3.1. Planen Sie das Netzwerk sorgfältig

Da NIS sensible Informationen unverschlüsselt über das Netzwerk überträgt, ist es wichtig, dass dieser Dienst hinter einer Firewall und auf einem segmentierten und sicheren Netzwerk ausgeführt wird. Jedes Mal, wenn NIS-Informationen über ein unsicheres Netzwerk übertragen werden, wird das Abfangen dieser Daten riskiert. Hier kann ein sorgfältiges Design des Netzwerks schwerwiegende

Sicherheitsbrüche verhindern.

2.2.3.2. Verwenden Sie passwortähnliche NIS-Domain-Namen und Hostnamen

Jede Maschine innerhalb einer NIS-Domain kann über bestimmte Befehle ohne Authentifizierung Informationen von einem Server abrufen, wenn der Benutzer den DNS-Hostnamen und den NIS-Domain-Namen des NIS-Servers kennt.

Wenn sich zum Beispiel jemand mit einem Laptop mit dem Netzwerk verbindet oder von außen ins Netzwerk eindringt (und es schafft, eine interne IP-Adresse vorzutäuschen), enthüllt der folgende Befehl die **/etc/passwd**-Map:

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

Ist der Angreifer ein Root-Benutzer, kann dieser die Datei **/etc/shadow** durch folgenden Befehl einsehen:

```
ypcat -d <NIS_domain> -h <DNS_hostname> shadow
```

Anmerkung

Wenn Kerberos verwendet wird, wird die Datei **/etc/shadow** nicht innerhalb einer NIS-Map gespeichert.

Um den Zugang zu NIS-Maps für einen Angreifer zu erschweren, erstellen Sie einen zufälligen String für den DNS-Hostnamen, wie zum Beispiel **o7hfawtgmhwg.domain.com**. Erstellen Sie auf die gleiche Weise einen *anderen*, zufallsgenerierten NIS-Domain-Namen. Hierdurch wird es einem Angreifer erheblich erschwert, Zugang zum NIS-Server zu erlangen.

2.2.3.3. Bearbeiten Sie die Datei /var/yp/securenets

NIS horcht auf alle Netzwerke, wenn die Datei **/var/yp/securenets** leer ist oder nicht existiert (dies ist z. B. nach einer Standardinstallation der Fall). Als Erstes sollten Sie ein Netzmaske/Netzwerkpaar in der Datei hinterlegen, damit **ypserv** nur auf Anfragen des richtigen Netzwerks reagiert.

Nachfolgend sehen Sie einen Beispieleintrag einer **/var/yp/securenets**-Datei:

```
255.255.255.0     192.168.0.0
```

Warnung

Sie sollten niemals einen NIS-Server zum ersten Mal starten, ohne vorher die Datei **/var/yp/securenets** erstellt zu haben.

Diese Methode schützt zwar nicht vor einer IP-Spoofing-Attacke, schränkt jedoch zumindest die Netzwerke ein, die vom NIS-Server bedient werden.

2.2.3.4. Weisen Sie statische Ports zu und nutzen Sie IPTables-Regeln

Jedem der zu NIS gehörenden Server kann ein bestimmter Port zugewiesen werden, mit Ausnahme von **rpc.yppasswdd** — dem Daemon, der Benutzern das Ändern ihrer Login-Passwörter erlaubt. Indem Sie

den anderen beiden NIS-Server-Daemons, **rpc.ypxfrd** und **ypserv**, Ports zuweisen, können Sie Firewall-Regeln erstellen, um die NIS-Server-Daemons noch mehr vor Angriffen zu schützen.

Fügen Sie dazu die folgenden Zeilen zu **/etc/sysconfig/network** hinzu:

```
YPSERV_ARGS="-p 834" YPXFRD_ARGS="-p 835"
```

Die folgenden IPTables-Regeln können dann verwendet werden, um festzulegen, auf welches Netzwerk der Server für diese Ports horchen soll:

```
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 834 -j DROP
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 835 -j DROP
```

Dies bedeutet, dass der Server nur Verbindungen zu den Ports 834 und 835 zulässt, wenn die Anfrage aus dem 192.168.0.0/24 Netzwerk kommt, unabhängig vom Protokoll.

Anmerkung

Siehe auch Abschnitt 2.5, „Firewalls“ für weitere Informationen zum Einrichten von Firewalls mit dem IPTables-Befehl.

2.2.3.5. Verwenden Sie Kerberos-Authentifizierung

Einer der größten Mängel beim Verwenden von NIS für Authentifizierung ist, dass ein Passwort-Hash der **/etc/shadow**-Map über das Netzwerk verschickt wird, sobald sich ein Benutzer an einem Computer anmeldet. Wenn ein Angreifer Zugang zu einer NIS-Domain erhält und Datenverkehr über das Netzwerk abfängt, können somit Benutzernamen und Passwort-Hashes unbemerkt gesammelt werden. Mit genügend Zeit kann dann ein Programm zum Knacken von Passwörtern schwache Passwörter ermitteln, wodurch ein Angreifer auf einen gültigen Account im Netzwerk zugreifen kann.

Da Kerberos Verschlüsselungen mit geheimen Schlüsseln einsetzt, werden niemals Passwort-Hashes über das Netzwerk versandt, was das System erheblich sicherer macht. Siehe *Managing Single Sign-On and Smart Cards* für weitere Informationen über Kerberos.

2.2.4. Sichern von NFS

Wichtig

Die Version von NFS, die Bestandteil von Red Hat Enterprise Linux 6 ist (NFSv4), benötigt nicht länger den **portmap**-Dienst, wie im Abschnitt 2.2.2, „Sichern von Portmap“ beschrieben. Der NFS-Datenverkehr benutzt statt UDP nunmehr TCP in allen Versionen und erfordert TCP bei der Verwendung von NFSv4. NFSv4 beinhaltet nun Kerberos Benutzer- und Gruppenauthentifizierung als Teil des **RPCSEC_GSS** Kernel-Moduls. Informationen über **portmap** sind jedoch nach wie vor enthalten, da Red Hat Enterprise Linux 6 auch noch NFSv2 und NFSv3 unterstützt, die **portmap** einsetzen.

2.2.4.1. Planen Sie das Netzwerk sorgfältig

Da nunmehr sämtliche Informationen von NFSv4 verschlüsselt mittels Kerberos über das Netzwerk übertragen werden können, ist es wichtig, dass dieser Dienst richtig konfiguriert wird, sollte sich dieser hinter einer Firewall oder in einem segmentierten Netzwerk befinden. NFSv2 und NFSv3 übergeben Daten dagegen nach wie vor nicht sicher, was unbedingt berücksichtigt werden sollte. Hier kann ein

sorgfältiges Design des Netzwerks schwerwiegende Sicherheitsbrüche verhindern.

2.2.4.2. Vermeiden Sie Syntaxfehler

Der NFS-Server entscheidet mithilfe der **`/etc/exports`**-Datei, welche Dateisysteme für welche Hosts exportiert werden sollen. Achten Sie darauf, dass Sie keine überflüssigen Leerstellen beim Bearbeiten dieser Datei einfügen.

Die folgende Zeile in der Datei **`/etc/exports`** legt fest, dass der Host **`bob.example.com`** Lese- und Schreibberechtigung auf das gemeinsam genutzte Verzeichnis **`/tmp/nfs/`** erhält.

```
/tmp/nfs/     bob.example.com(rw)
```

Folgende Zeile in der Datei **`/etc/exports`** legt dagegen fest, dass der Host **`bob.example.com`** lediglich Leseberechtigung besitzt, allerdings *jeder andere* Host Lese- und Schreibberechtigung hat, und das wegen eines einzelnen Leerzeichens nach dem Hostnamen.

```
/tmp/nfs/     bob.example.com (rw)
```

Es ist sehr sinnvoll, alle konfigurierten NFS-Shares mit dem **`showmount`**-Befehl zu prüfen:

```
showmount -e <hostname>
```

2.2.4.3. Verwenden Sie nicht die Option no_root_squash

Standardmäßig ändern NFS-Shares den Root-Benutzer in den Benutzer **`nfsnobody`** um, einen unprivilegierten Benutzer-Account. Auf diese Weise gehören alle von Root erstellten Dateien dem Benutzer **`nfsnobody`**, wodurch das Laden von Programmen mit gesetztem Setuid-Bit verhindert wird.

Wenn jedoch **`no_root_squash`** verwendet wird, können Remote-Root-Benutzer jede Datei in dem gemeinsamen Dateisystem verändern und dabei mit Trojanern infizierte Anwendungen hinterlassen, die von anderen Benutzern unbeabsichtigt ausgeführt werden.

2.2.4.4. NFS Firewall-Konfiguration

Die für NFS verwendeten Ports werden dynamisch von rpcbind zugewiesen, was zu Schwierigkeiten bei der Konfiguration von Firewall-Regeln führen kann. Um diesen Prozess zu vereinfachen, verwenden Sie die */etc/sysconfig/nfs*-Datei, um die zu verwendenden Ports festzulegen:

- **`MOUNTD_PORT`** — TCP und UDP Port für mountd (rpc.mountd)
- **`STATD_PORT`** — TCP und UDP Port für status (rpc.statd)
- **`LOCKD_TCPPORT`** — TCP Port für nlockmgr (rpc.lockd)
- **`LOCKD_UDPPORT`** — UDP Port für nlockmgr (rpc.lockd)

Die spezifizierten Port-Nummern dürfen von keinem anderen Dienst verwendet werden. Konfigurieren Sie anschließend Ihre Firewall, um die gewählten Port-Nummern sowie TCP und UDP Port 2049 (NFS) zu erlauben.

Führen Sie den Befehl **`rpcinfo -p`** auf dem NFS-Server aus um zu überprüfen, welche Ports und RPC-Programme verwendet werden.

2.2.5. Sicherung des Apache HTTP-Server

Der Apache HTTP-Server ist einer der stabilsten und sichersten Dienste, die mit Red Hat Enterprise Linux ausgeliefert werden. Es gibt eine große Anzahl von Optionen und Methoden, um den Apache

HTTP-Server zu sichern — zu viele, um sie hier im Detail zu beschreiben. Der folgende Abschnitt geht kurz auf die empfohlenen Verfahren beim Einsatz von Apache HTTP-Server ein.

Vergewissern Sie sich grundsätzlich, dass jegliche Skripte auf dem System auch wie beabsichtigt funktionieren, *bevor* sie in Produktion gegeben werden. Stellen Sie außerdem sicher, dass nur der Root-Benutzer Schreibberechtigungen für Verzeichnisse besitzt, die Skripte oder CGIs enthalten. Führen Sie dazu die folgenden Befehle als Root-Benutzer aus:

1. ```
   chown root <directory_name>
   ```

2. ```
   chmod 755 <directory_name>
   ```

Systemadministratoren sollten folgende Konfigurationsoptionen mit äußerster Sorgfalt verwenden (konfiguriert in **`/etc/httpd/conf/httpd.conf`**):

`FollowSymLinks`
: Diese Direktive ist standardmäßig aktiviert, seien Sie also vorsichtig, wenn Sie symbolische Links zum Document-Root des Webservers erstellen. Es ist zum Beispiel keine gute Idee, einen symbolischen Link zu **`/`** anzugeben.

`Indexes`
: Diese Direktive ist standardmäßig aktiviert, ist jedoch unter Umständen nicht wünschenswert. Wenn Sie nicht möchten, dass Benutzer Dateien auf dem Server durchsuchen, ist es sinnvoll, diese Direktive zu entfernen.

`UserDir`
: Die **`UserDir`**-Direktive ist standardmäßig deaktiviert, da sie das Vorhandensein eines Benutzer-Accounts im System bestätigen kann. Wenn Sie das Durchsuchen von Verzeichnissen auf dem Server durch Benutzer erlauben möchten, sollten Sie die folgenden Direktiven verwenden:

```
UserDir enabled
UserDir disabled root
```

Diese Direktiven aktivieren das Durchsuchen von Verzeichnissen für alle Benutzerverzeichnisse außer **`/root`**. Wenn Sie Benutzer zu der Liste deaktivierter Accounts hinzufügen möchten, können Sie eine durch Leerstellen getrennte Liste der Benutzer in die Zeile **`UserDir disabled`** einfügen.

Wichtig

Entfernen Sie nicht die **`IncludesNoExec`**-Direktive. Standardmäßig kann das Modul *Server-Side Includes* (SSI) keine Befehle ausführen. Es wird davon abgeraten, diese Einstellungen zu ändern, außer wenn unbedingt notwendig, da dies einem Angreifer ermöglichen könnte, Befehle auf dem System auszuführen.

2.2.6. Sichern von FTP

Das *File Transport Protocol* (*FTP*) ist ein älteres TCP-Protokoll, das zum Übertragen von Dateien über ein Netzwerk entwickelt wurde. Da alle Transaktionen mit dem Server, einschließlich der Benutzerauthentifizierung, unverschlüsselt ablaufen, gilt es als unsicheres Protokoll und sollte sorgfältig konfiguriert werden.

Red Hat Enterprise Linux bietet drei FTP-Server.

- **gssftpd** — Ein Kerberos-fähiger, **xinetd**-basierter FTP-Daemon, der keine Authentifizierungsinformationen über das Netzwerk überträgt.
- **Red Hat Content Accelerator** (**tux**) — Ein Kernel-Space Webserver mit FTP-Fähigkeiten.
- **vsftpd** — Eine eigenständige, sicherheitsorientierte Implementierung des FTP-Dienstes.

Die folgenden Sicherheitsrichtlinien gelten für das Einrichten des **vsftpd**-FTP-Dienstes.

2.2.6.1. FTP-Grußbanner

Bevor der Benutzername und das Passwort eingereicht werden, erhalten alle Benutzer ein Grußbanner. Standardmäßig enthält dieses Banner Versionsinformationen, die für Cracker nützlich sein können, die Schwachstellen in einem System herausfinden wollen.

Um dieses Grußbanner für **vsftpd** zu ändern, fügen Sie die folgende Direktive zu **/etc/vsftpd/vsftpd.conf**-Datei hinzu:

```
ftpd_banner=<insert_greeting_here>
```

Ersetzen Sie ***<insert_greeting_here>*** in der obigen Direktive durch den Text Ihrer Begrüßung.

Für mehrzeilige Banner ist es ratsam, eine Bannerdatei zu verwenden. Um die Verwaltung von mehreren Bannern zu vereinfachen, speichern Sie alle Banner in einem neuen Verzeichnis namens **/etc/banners/**. Die Bannerdatei für FTP-Verbindungen in diesem Beispiel ist **/etc/banners/ftp.msg**. Das nachfolgende Beispiel zeigt, wie eine derartige Datei aussehen kann:

```
######### # Hello, all activity on ftp.example.com is logged. #########
```

Anmerkung

Es ist nicht nötig, jede Zeile der Datei mit **220**, wie in Abschnitt 2.2.1.1.1, „TCP-Wrapper und Verbindungsbanner“ beschrieben, zu beginnen.

Um für **vsftpd** auf diese Grußbanner-Datei zu verweisen, fügen Sie folgende Direktive zu **/etc/vsftpd/vsftpd.conf** hinzu:

```
banner_file=/etc/banners/ftp.msg
```

Es ist auch möglich, zusätzliche Banner für eingehende Verbindungen mittels TCP-Wrappern zu senden. Dies wird unter Abschnitt 2.2.1.1.1, „TCP-Wrapper und Verbindungsbanner“ beschrieben.

2.2.6.2. Anonymer Zugang

Die Existenz des **/var/ftp/**-Verzeichnisses aktiviert den anonymen Account.

Der einfachste Weg, dieses Verzeichnis zu erstellen, ist durch die Installation des **vsftpd**-Pakets. Dieses Paket erstellt einen Verzeichnisbaum für anonyme Benutzer und vergibt anonymen Benutzern

lediglich Leseberechtigungen für Verzeichnisse.

Standardmäßig können anonyme Benutzer nicht in Verzeichnisse schreiben.

Warnung

Wenn Sie einen anonymen Zugang zu FTP-Servern zulassen, sollten Sie darauf achten, wo Sie sensible Daten speichern.

2.2.6.2.1. Anonymes Hochladen

Wenn Sie anonymen Benutzern erlauben möchten, Dateien hochzuladen, wird empfohlen, ein Verzeichnis nur mit Schreibberechtigung innerhalb von**/var/ftp/pub/** anzulegen.

Führen Sie dazu den folgenden Befehl aus:

```
mkdir /var/ftp/pub/upload
```

Ändern Sie dann wie folgt die Berechtigungen, so dass anonyme Benutzer nicht sehen können, was sich innerhalb des Verzeichnisses befindet:

```
chmod 730 /var/ftp/pub/upload
```

Ein detailliertes Listing des Verzeichnisses sollte wie folgt aussehen:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

Warnung

Administratoren, die anonymen Benutzern Lese- und Schreibberechtigungen für Verzeichnisse geben, stellen häufig fest, dass ihr Server dann zu einer Fundgrube gestohlener Software wird.

Fügen Sie zusätzlich unter **vsftpd** die folgende Zeile in die Datei **/etc/vsftpd/vsftpd.conf** ein:

```
anon_upload_enable=YES
```

2.2.6.3. Benutzer-Accounts

Da FTP Benutzernamen und Passwörter unverschlüsselt über unsichere Netzwerke zur Authentifizierung überträgt, ist es ratsam, Systembenutzern den Zugang zum Server von ihren Benutzer-Accounts aus zu verbieten.

Um alle Benutzer-Accounts in **vsftpd** zu deaktivieren, fügen Sie die folgende Direktive zu **/etc/vsftpd/vsftpd.conf** hinzu:

```
local_enable=NO
```

2.2.6.3.1. Einschränken von Benutzer-Accounts

Der einfachste Weg, eine bestimmte Gruppe von Accounts, wie den Root-Benutzer und solche mit **sudo**-Berechtigungen, am Zugriff auf den FTP-Server zu hindern, ist durch eine PAM-Listendatei, wie unter

Abschnitt 2.1.4.2.4, „Deaktivieren von PAM für Root“ beschrieben. Die PAM-Konfigurationsdatei für **vsftpd** ist **/etc/pam.d/vsftpd**.

Es ist auch möglich, Benutzer-Accounts direkt innerhalb einzelner Dienste zu deaktivieren.

Um bestimmte Benutzer-Accounts in **vsftpd** zu deaktivieren, fügen Sie den Benutzernamen zu **/etc/vsftpd/ftpusers** hinzu.

2.2.6.4. TCP-Wrapper für die Zugriffskontrolle

Sie können TCP-Wrapper für die Zugriffskontrolle zu den FTP-Daemons wie unter Abschnitt 2.2.1.1, „Erhöhung der Sicherheit mit TCP-Wrappern“ beschrieben einsetzen.

2.2.7. Sichern von Sendmail

Sendmail ist ein Mail Transport Agent (MTA), der das Simple Mail Transport Protocol (SMTP) zur Übertragung elektronischer Nachrichten zwischen anderen MTAs und für das E-Mailen an Clients oder Delivery Agents einsetzt. Obwohl viele MTAs den Verkehr untereinander verschlüsseln können, tun dies viele nicht, so dass das Versenden von E-Mails über ein öffentliches Netzwerk als eine von Natur aus unsichere Form der Kommunikation gilt.

Es wird empfohlen, dass Sie sich mit den folgenden Themen auseinandersetzen, wenn Sie die Implementierung eines Sendmail-Servers planen.

2.2.7.1. Einschränken von Denial-of-Service-Angriffen

Aufgrund der Beschaffenheit von E-Mail kann ein dazu entschlossener Angreifer den Server leicht mit E-Mails überfluten und so ein Denial-of-Service verursachen. Indem Sie in die folgenden Direktiven in **/etc/mail/sendmail.mc** mit Grenzwerten versehen, kann die Wirksamkeit solcher Angriffe stark abgeschwächt werden.

- **confCONNECTION_RATE_THROTTLE** — Die Anzahl der Verbindungen, die der Server pro Sekunde empfangen kann. Standardmäßig begrenzt Sendmail die Zahl der Verbindungen nicht. Wird eine Grenze gesetzt, werden darüber hinaus gehende Verbindungen verzögert.
- **confMAX_DAEMON_CHILDREN** — Die maximale Anzahl von untergeordneten Prozessen, die vom Server erzeugt werden können. Standardmäßig begrenzt Sendmail die Anzahl der untergeordneten Prozesse nicht. Wird eine Grenze gesetzt, werden alle darüber hinaus gehenden Verbindungen verzögert.
- **confMIN_FREE_BLOCKS** — Die minimale Anzahl freier Blöcke, die für den Server zur Verfügung stehen müssen, um E-Mail empfangen zu können. Der Standard beträgt 100 Blöcke.
- **confMAX_HEADERS_LENGTH** — Die maximal akzeptierte Größe (in Bytes) für einen Nachrichten-Header.
- **confMAX_MESSAGE_SIZE** — Die maximal akzeptierte Größe (in Bytes) pro Nachricht.

2.2.7.2. NFS und Sendmail

Legen Sie niemals das Mail-Spool-Verzeichnis, **/var/spool/mail/**, auf einem durch NFS gemeinsam genutzten Datenträger ab.

Da NFSv2 und NFSv3 keine Kontrolle über Benutzer- und Gruppen-IDs haben, können zwei oder mehr Benutzer die gleiche UID besitzen und daher jeweils die E-Mails des anderen lesen.

Anmerkung

Mit NFSv4 und Kerberos ist dies nicht der Fall, da das **SECRPC_GSS**-Kernel-Modul keine UID-basierte Authentifizierung anwendet. Allerdings sollten Sie dennoch das Mail-Spool-Verzeichnis *nicht* auf einem durch NFS gemeinsam genutzten Datenträger ablegen.

2.2.7.3. Nur-Mail Benutzer

Um Sicherheitslücken des Sendmail-Servers bei lokalen Benutzern zu vermeiden, ist es am besten, wenn Mail-Benutzer nur über ein E-Mail-Programm auf den Sendmail-Server zugreifen. Shell-Accounts auf dem Mail-Server sollten nicht erlaubt sein, und alle Benutzer-Shells in der Datei **/etc/passwd** sollten auf **/sbin/nologin** gesetzt sein (evtl. unter Ausnahme des Root-Benutzers).

2.2.8. Überprüfen der horchenden Ports

Nachdem Sie die Netzwerkdienste konfiguriert haben, ist es wichtig zu überprüfen, welche Ports auf die Netzwerkschnittstellen im System horchen. Etwaige offene Ports können Beweis für ein unbefugtes Eindringen sein.

Es gibt zwei grundlegende Herangehensweisen für das Auflisten der Ports, die auf das Netzwerk horchen. Die weniger zuverlässige Methode ist, den Netzwerkstapel durch Befehle wie **netstat -an** oder **lsof -i** abzufragen. Diese Methode ist deshalb unzuverlässiger, da derartige Programme sich nicht vom Netzwerk aus mit dem Computer verbinden, sondern vielmehr prüfen, was auf dem System ausgeführt wird. Aus diesen Grund sind diese Anwendungen häufig Ziel für Ersetzungen durch Angreifer. Bei dieser Methode versuchen Cracker, ihre Spuren zu verwischen, wenn diese unbefugt Netzwerkports geöffnet haben, indem sie die Anwendungen **netstat** und **lsof** durch ihre eigenen, modifizierten Versionen ersetzen.

Ein zuverlässigerer Weg zum Überprüfen, welche Ports auf das Netzwerk horchen, ist die Verwendung eines Port-Scanners wie z. B. **nmap**.

Der folgende Befehl, von einer Konsole aus eingegeben, stellt fest, welche Ports auf TCP-Verbindungen aus dem Netzwerk horchen.

```
nmap -sT -O localhost
```

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar  2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

Diese Ausgabe zeigt, dass das System **portmap** ausführt, da der Dienst **sunrpc** vorhanden ist. Es wird jedoch auch ein unbekannter Dienst auf Port 834 ausgeführt. Um zu prüfen, ob dieser Port zu der offiziellen Liste bekannter Dienste gehört, geben Sie Folgendes ein:

```
cat /etc/services | grep 834
```

Dieser Befehl liefert keine Ausgabe für Port 834. Aufgrund des Befehlsformats wird jedoch Ausgabe für andere Ports (1834, 2834 und 3834) angezeigt. Dies bedeutet, dass der Port 834 im reservierten Bereich (0 bis 1023) liegt und Root-Zugang zum Öffnen benötigt, jedoch nicht mit einem bekannten Dienste zusammenhängt.

Rufen Sie als Nächstes mittels **netstat** oder **lsof** Informationen über den Port ab. Um Port 834 mithilfe von **netstat** zu prüfen, geben Sie folgenden Befehl ein:

```
netstat -anp | grep 834
```

Dieser Befehl liefert folgende Ausgabe:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

Dass der offene Port in **netstat** aufgeführt wird, ist ein gutes Zeichen, da ein Cracker, der einen Port heimlich auf einem geknackten System öffnet, das Anzeigen des Ports durch diesen Befehl höchstwahrscheinlich nicht zulassen würde. Des Weiteren zeigt die Option **[p]** die Prozess-ID (PID) des Dienstes an, der diesen Port geöffnet hat. In diesem Fall gehört der offene Port zu **ypbind** (NIS), ein RPC-Dienst, der zusammen mit dem **portmap**-Dienst läuft.

Der **lsof**-Befehl zeigt ähnliche Informationen wie der **netstat**-Befehl an, denn er kann offene Ports auch Diensten zuordnen:

```
lsof -i | grep 834
```

Der relevante Teil der Befehlsausgabe ist der folgende Abschnitt:

```
ypbind      653        0    7u  IPv4        1319                  TCP *:834
(LISTEN)
ypbind      655        0    7u  IPv4        1319                  TCP *:834
(LISTEN)
ypbind      656        0    7u  IPv4        1319                  TCP *:834
(LISTEN)
ypbind      657        0    7u  IPv4        1319                  TCP *:834
(LISTEN)
```

Wie Sie sehen, können diese Tools eine Menge Informationen über den Status von Diensten auf einem Computer ausgeben. Diese Tools sind flexibel und liefern eine Vielzahl von Informationen über die Netzwerkdienste und zur Konfiguration. Werfen Sie einen Blick auf die Handbuchseiten von **lsof**, **netstat**, **nmap** und **services** für weitere Informationen.

2.3. TCP-Wrapper und xinetd

Die Kontrolle über den Zugriff auf Netzwerkdienste ist eine der wichtigsten Sicherheitsaufgaben, denen sich ein Server-Administrator stellen muss. Unter Red Hat Enterprise Linux gibt es eine Reihe von Tools zu diesem Zweck. Eine **iptables**-basierte Firewall etwa filtert alle unerwünschten Netzwerkpakete im Netzwerkstapel des Kernels heraus. Für Netzwerkdienste, die davon Gebrauch machen, fügt *TCP Wrapper* eine zusätzliche Schutzschicht hinzu, indem dieser definiert, welchen Hosts es erlaubt ist mit Netzwerkdiensten zu verbinden, die von TCP Wrappern geschützt werden, und welchen nicht. Einer dieser durch TCP Wrapper geschützten Netzwerkdienste ist der **xinetd** *Super-Server*. Dieser Dienst wird Super-Server genannt, da er Verbindungen zu einer Untergruppe von Netzwerkdiensten steuert und die Zugriffskontrolle weiter verfeinert.

Abbildung 2.4, „Zugriffskontrolle zu Netzwerkdiensten“ ist eine einfache Illustration, die die Zusammenarbeit dieser Tools beim Schutz von Netzwerkdiensten darstellt.

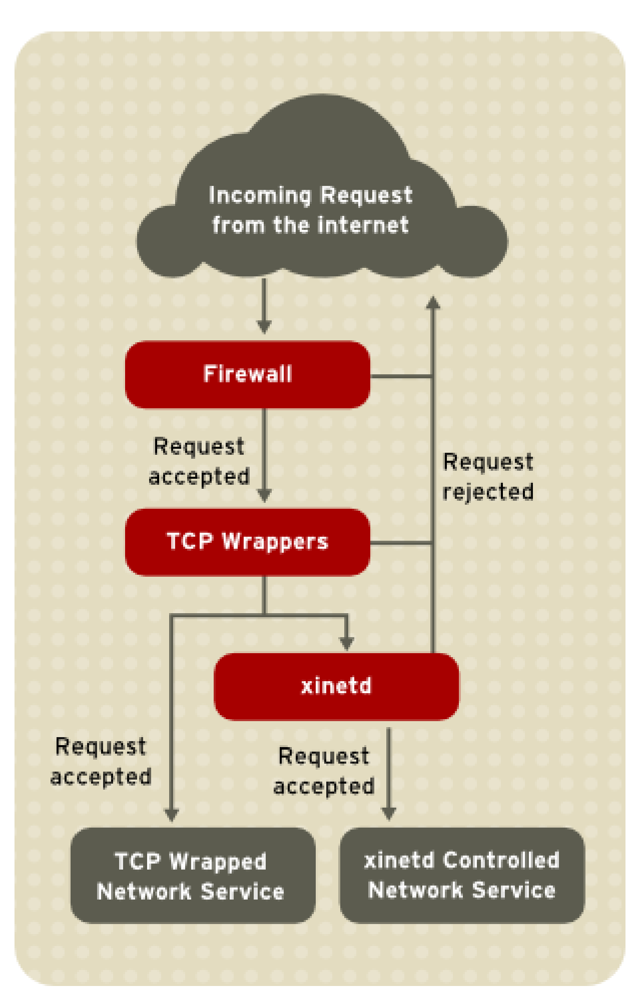

Abbildung 2.4. Zugriffskontrolle zu Netzwerkdiensten

Dieses Kapitel beschäftigt sich mit der Rolle von TCP-Wrappern und **`xinetd`** bei der Zugriffskontrolle auf Netzwerkdienste sowie mit Wegen, wie mithilfe dieser Tools die Verwaltung der Protokollierung und der Anwendung verbessern werden kann. Weitere Informationen zum Einsatz von Firewalls mit **`iptables`** finden Sie unter Abschnitt 2.6, „IPTables“.

2.3.1. TCP Wrappers

Die TCP-Wrapper-Pakete (**`tcp_wrappers`** und **`tcp_wrappers-libs`**) sind standardmäßig installiert und stellen Host-basierte Zugriffskontrolle für Netzwerkdienste zur Verfügung. Die wichtigste Komponente in diesen Paketen ist die **`/lib/libwrap.a`** oder **`/lib64/libwrap.a`**-Bibliothek. Im Wesentlichen handelt es sich bei einem von TCP Wrappern kontrollierten Dienst um einen Dienst, der mit der **`libwrap.a`**-Bibliothek kompiliert wurde.

Wenn ein Verbindungsversuch zu einem TCP-wrapped Dienst eingeleitet wird, so wird der Dienst zuerst die Zugriffsdateien des Hosts (**`/etc/hosts.allow`** und **`/etc/hosts.deny`**) untersuchen um festzustellen, ob eine Verbindung des Clients erlaubt ist. In den meisten Fällen schreibt er anschließend mithilfe des syslog-Daemons (**`syslogd`**) den Namen des anfordernden Hosts und Dienstes in **`/var/log/secure`** oder **`/var/log/messages`**.

Wenn es einem Client erlaubt ist sich zu verbinden, gibt der TCP-Wrapper die Kontrolle über die Verbindung an den angeforderten Dienst ab und greift nicht weiter in die Kommunikation zwischen Client und Server ein.

Zusätzlich zu Zugriffskontrolle und Protokollierung können TCP-Wrapper Befehle ausführen, um mit

dem Client zu interagieren, ehe die Kontrolle der Verbindung zum angeforderten Netzwerkdienst übergeben oder abgelehnt wird.

Da TCP-Wrapper eine wertvolle Ergänzung im Arsenal von Sicherheitstools eines jeden Systemadministrators sind, sind die meisten Netzwerkdienste unter Red Hat Enterprise Linux mit der **`libwrap.a`**-Bibliothek verbunden. Zu diesen Applikationen gehören **`/usr/sbin/sshd`**, **`/usr/sbin/sendmail`** und **`/usr/sbin/xinetd`**.

Anmerkung

Um festzustellen, ob die Binärdatei eines Netzwerkdienstes mit **`libwrap.a`** verknüpft ist, geben Sie folgenden Befehl als Root-Benutzer ein:

```
ldd <binary-name> | grep libwrap
```

Ersetzen Sie ***`<binary-name>`*** dabei durch dem Namen der Binärdatei des Netzwerkdienstes. Falls der Befehl ohne Ausgabe direkt zur Befehlszeile zurückkehrt, so ist der Netzwerkdienst *nicht* mit **`libwrap.a`** verknüpft.
Das folgende Beispiel zeigt, dass **`/usr/sbin/sshd`** mit **`libwrap.a`** verknüpft ist:

```
[root@myServer ~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@myServer ~]#
```

2.3.1.1. Vorteile von TCP-Wrappern

TCP-Wrapper bieten im Vergleich zu anderen Kontrollmethoden für Netzwerkdienste die folgenden Vorteile:

- *Transparenz für sowohl Client als auch den TCP-wrapped Netzwerkdienst* — Weder der sich verbindende Client noch der wrapped Netzwerkdienst merken, dass TCP-Wrapper in Einsatz sind. Verbindungsversuche von berechtigten Benutzern werden protokolliert und mit dem geforderten Dienst verbunden, während Verbindungsversuche unzulässiger Clients fehlschlagen.
- *Zentralisierte Verwaltung mehrerer Protokolle* — TCP-Wrapper arbeiten unabhängig von den Netzwerkdiensten, die sie schützen. Dadurch können sich mehrere Server-Applikationen einen gemeinsamen Satz von Konfigurationsdateien der Zugriffskontrolle teilen, was die Verwaltung vereinfacht.

2.3.2. TCP-Wrapper Konfigurationsdateien

Um festzustellen, ob es einem Client erlaubt ist, sich mit einem bestimmten Dienst zu verbinden, verwenden TCP Wrapper die folgenden beiden Dateien, die auch als *hosts access*-Dateien bezeichnet werden:

- **`/etc/hosts.allow`**
- **`/etc/hosts.deny`**

Wenn bei einem TCP-wrapped Dienst eine Client-Anfrage eingeht, führt er die folgenden Schritte durch:

1. *Er referenziert **`/etc/hosts.allow`*** — Der TCP-wrapped Dienst analysiert die **`/etc/hosts.allow`**-Datei sequentiell und wendet die erste Regel an, die für diesen Dienst festgelegt wurde. Wenn eine passende Regel ausfindig gemacht werden kann, erlaubt der Dienst

die Verbindung. Wenn nicht, geht er zum nächsten Schritt über.

2. *Er referenziert* ***/etc/hosts.deny*** — Der TCP-wrapped Dienst analysiert die **/etc/hosts.deny**-Datei sequentiell. Wenn eine passende Regel ausfindig gemacht werden kann, lehnt der Dienst die Verbindung ab. Wenn nicht, wird der Zugang zu diesem Dienst bewilligt.

Die folgenden Punkte sind wichtig, wenn TCP-Wrapper verwendet werden, um Netzwerkdienste zu schützen:

- Da Zugriffsregeln in **hosts.allow** zuerst angewendet werden, haben diese Vorrang vor den Regeln in **hosts.deny**. Sollte der Zugriff zu einem Dienst in **hosts.allow** erlaubt sein, so wird eine den Zugriff auf diesen Dienst verbietende Regel in **hosts.deny** ignoriert.
- Da alle Regeln von oben nach unten abgearbeitet werden, wird lediglich die erste auf einen Dienst passende Regel angewendet, weshalb die Reihenfolge der Regeln extrem wichtig ist.
- Sollte keine Regel für den Dienst gefunden werden oder keine der beiden Dateien vorhanden sein, so wird der Zugriff zu diesem Dienst gewährt.
- TCP-wrapped Dienste speichern Regeln für die Hosts-Zugriffsdateien nicht zwischen. Jegliche Änderungen an **hosts.allow** oder **hosts.deny** treten daher auch ohne Neustart der Netzwerkdienste sofort in Kraft.

Warnung

Sollte die letzte Zeile einer Hosts-Zugriffsdatei kein Zeilenvorschubzeichen sein (durch Drücken der **Enter**-Taste erzeugt), schlägt die letzte Regel in der Datei fehl und ein Fehler wird entweder in **/var/log/messages** oder **/var/log/secure** protokolliert. Dies ist auch der Fall für Regeln, die ohne Backslash-Zeichen auf mehrere Zeilen umgebrochen sind. Das folgende Beispiel zeigt den relevanten Teil einer Protokollmeldung für eine durch genannte Gründe fehlerhafte Regel:

```
warning: /etc/hosts.allow, line 20: missing newline or line too long
```

2.3.2.1. Formatierung von Zugriffsregeln

Das Format der beiden Dateien **/etc/hosts.allow** and **/etc/hosts.deny** ist identisch. Jede Regel muss in einer neuen Zeile beginnen. Leere Zeilen oder Zeilen, die mit dem Rautenzeichen (#) beginnen, werden ignoriert.

Jede Regel verwendet das folgende, grundlegende Format, um den Zugriff zu Netzwerkdiensten zu steuern:

```
<daemon list>: <client list> [: <option>: <option>: ...]
```

- ***<daemon list>*** — Eine kommagetrennte Liste mit Prozessnamen (*nicht* Dienstnamen) oder der **ALL**-Platzhalter. Die Daemon-Liste akzeptiert auch Operatoren (siehe Abschnitt 2.3.2.1.4, „Operatoren“) für größere Flexibilität.
- ***<client list>*** — Eine kommagetrennte Liste mit Hostnamen, Host-IP-Adressen, bestimmten Zeichenketten oder Platzhaltern, die die von der Regel betroffenen Hosts spezifizieren. Die Client-Liste akzeptiert auch Operatoren (siehe Abschnitt 2.3.2.1.4, „Operatoren“) für größere Flexibilität.
- ***<option>*** — Eine optionale Aktion oder durch Doppelpunkte getrennte Liste von Aktionen, die ausgeführt werden, wenn eine Regel angewendet wird. Optionsfelder unterstützen Expansionen, führen Shell-Befehle aus, gewähren Zugriff oder lehnen diesen ab, und ändern das Protokollierungsverhalten.

Anmerkung

Weitere Informationen zu den oben erwähnten Begriffen finden Sie an anderen Stellen in diesem Handbuch:

- Abschnitt 2.3.2.1.1, „Platzhalter“
- Abschnitt 2.3.2.1.2, „Muster“
- Abschnitt 2.3.2.2.4, „Erweiterungen“
- Abschnitt 2.3.2.2, „Optionsfelder“

Nachfolgend sehen Sie ein einfaches Beispiel für eine Hosts-Zugriffsregel:

```
vsftpd : .example.com
```

Diese Regel weist TCP-Wrapper an, nach Verbindungen zum FTP-Daemon (**vsftpd**) von jedem Host in der **example.com**-Domain Ausschau zu halten. Wird diese Regel in **hosts.allow** eingefügt, so wird die Verbindung angenommen. Wird diese Regel dagegen in **hosts.deny** eingefügt, so wird die Verbindung abgelehnt.

Folgendes Beispiel einer Hosts-Zugriffsregel ist komplizierter und verwendet zwei Optionsfelder:

```
sshd : .example.com  \ : spawn /bin/echo `/bin/date` access
denied>>/var/log/sshd.log \ : deny
```

Beachten Sie, dass in diesem Beispiel jedem der Optionsfelder ein Backslash (\) vorausgeht. Die Verwendung eines Backslash verhindert, dass eine Regel aufgrund ihrer Länge nicht funktioniert.

Diese Beispielregel besagt, dass bei einem Verbindungsversuch zum SSH-Daemon (**sshd**) von einem Host in der **example.com**-Domain der **echo**-Befehl ausgeführt wird (der den Verbindungsversuch in eine spezielle Protokolldatei schreibt) und die Verbindung abgelehnt wird. Da die optionale **deny**-Direktive verwendet wird, wird diese Zeile den Zugriff ablehnen, auch wenn sie in der **hosts.allow**-Datei erscheint. Für einen detaillierteren Überblick der Optionen, siehe Abschnitt 2.3.2.2, „Optionsfelder“.

2.3.2.1.1. Platzhalter

Platzhalter erlauben TCP-Wrappern eine einfachere Übereinstimmung mit Gruppen von Daemons oder Hosts. Platzhalter werden häufig im Client-Listenfeld der Zugriffsregeln verwendet.

Die folgenden Platzhalter stehen zur Verfügung:

- **ALL** — Stimmt mit allen Werten überein. Kann sowohl für die Daemon-Liste als auch für die Client-Liste verwendet werden.
- **LOCAL** — Stimmt mit jedem Host überein, der keinen Punkt (.) enthält, wie z. B. localhost.
- **KNOWN** — Stimmt mit jedem Host überein, dessen Host-Name und Host-Adresse oder der Benutzer bekannt sind.
- **UNKNOWN** — Stimmt mit jedem Host überein, dessen Host-Name und Host-Adresse oder der Benutzer unbekannt sind.
- **PARANOID** — Stimmt mit jedem Host überein, dessen Host-Name nicht mit der Host-Adresse übereinstimmt.

Wichtig

Die Platzhalter **`KNOWN`**, **`UNKNOWN`** und **`PARANOID`** sollten mit Vorsicht verwendet werden, da deren ordnungsgemäßer Betrieb von einem funktionierenden DNS-Server abhängt. Ein Problem bei der Namensauflösung kann eine Zugriffsverweigerung auf Dienste für berechtigte Benutzer zur Folge haben.

2.3.2.1.2. Muster

Muster können im Client-Listenfeld von Zugriffsregeln benutzt werden, um Gruppen von Client-Hosts genauer zu bestimmen.

Nachfolgend sehen Sie eine Liste der gängigsten Muster für Einträge in der Client-Liste:

- *Hostname beginnt mit einem Punkt (.)* — Ein Punkt am Anfang eines Host-Namens bewirkt, dass auf alle Host-Rechner, die in diesem Hostnamen enden, die Regel angewendet wird. Das folgende Beispiel trifft auf jeden Host in der **`example.com`** Domain zu:

```
ALL : .example.com
```

- *IP-Adresse endet mit einem Punkt (.)* — Ein Punkt am Ende einer IP-Adresse bewirkt, dass auf alle Hosts, deren IP-Adresse mit derselben numerischen Gruppe beginnt, die Regel angewendet wird. Das folgende Beispiel trifft auf jeden Host im **`192.168.x.x`**-Netzwerk zu:

```
ALL : 192.168.
```

- *IP-Adresse/Netzmaske-Paar* — Netzmasken-Ausdrücke können auch als ein Muster verwendet werden, um den Zugriff zu einer bestimmten Gruppe von IP-Adressen zu regeln. Das folgende Beispiel trifft auf alle Hosts mit einer Adresse zwischen **`192.168.0.0`** und **`192.168.1.255`** zu:

```
ALL : 192.168.0.0/255.255.254.0
```

Wichtig

Wenn im IPv4-Adressraum gearbeitet wird, werden paarweise Deklarationen von Adresse/Präfixlänge (*prefixlen*) (CIDR-Notation) nicht unterstützt. Lediglich IPv6-Regeln können dieses Format verwenden.

- *[IPv6 Adresse]/prefixlen Paar* — [net]/prefixlen Paare können auch als Muster verwendet werden, um den Zugriff zu einer bestimmten Gruppe von IPv6-Adressen zu regeln. Das folgende Beispiel trifft auf jeden Host mit einem Adressbereich von **`3ffe:505:2:1::`** bis **`3ffe:505:2:1:ffff:ffff:ffff:ffff`** zu:

```
ALL : [3ffe:505:2:1::]/64
```

- *Ein Sternchen (*)* — Sternchen können für komplette Gruppen von Host-Namen oder IP-Adressen verwendet werden, solange diese nicht in einer Client-Liste verwendet werden, die bereits andere Arten von Muster verwendet. Das folgende Beispiel trifft auf alle Hosts in der **`example.com`**-Domain zu:

```
ALL : *.example.com
```

- *Der Schrägstrich (/)* — Wenn die Client-Liste mit einem Schrägstrich beginnt, wird diese als Dateiname behandelt. Dies ist nützlich, wenn Regeln benötigt werden, die eine große Anzahl von Hosts angeben. Das folgende Beispiel verweist TCP-Wrapper auf die **/etc/telnet.hosts**-Datei für alle Telnet-Verbindungen:

```
in.telnetd : /etc/telnet.hosts
```

Es gibt noch weitere, weniger häufig verwendete Muster, die ebenfalls von TCP-Wrappern akzeptiert werden. Weitere Informationen finden Sie auf der **hosts_access**(5)-Handbuchseite.

Warnung

Seien Sie sehr vorsichtig bei der Verwendung von Host- und Domain-Namen. Ein Angreifer kann verschiedene Tricks anwenden, um die richtige Namensauflösung zu umgehen. Zudem hindert ein Ausfall des DNS-Dienstes sogar berechtigte Benutzer an der Verwendung von Netzwerkdiensten. Es sollten daher wann immer möglich IP-Adressen verwendet werden.

2.3.2.1.3. Portmap und TCP Wrappers

Portmaps Implementierung von TCP-Wrappern unterstützt keine Namensauflösung, was bedeutet, dass **portmap** keine Host-Namen zur Identifizierung von Hosts verwenden kann. Daher müssen Regeln für die Zugriffskontrolle für Portmap in **hosts.allow** oder **hosts.deny** IP-Adressen oder den Schlüsselbegriff **ALL** für die Spezifizierung von Hosts verwenden.

Änderungen an den **portmap**-Zugriffskontrollregeln werden nicht sofort wirksam. Sie müssen ggf. den **portmap**-Dienst neu starten.

Da der Betrieb von weit verbreiteten Diensten wie NIS und NFS von **portmap** abhängt, bedenken Sie diese Einschränkungen.

2.3.2.1.4. Operatoren

Die Zugriffskontrollregeln akzeptieren derzeit einen Operator, **EXCEPT**. Dieser kann sowohl in der Daemon- als auch in der Client-Liste einer Regel verwendet werden.

Der **EXCEPT**-Operator erlaubt spezifische Ausnahmen an breiter gefächerten Treffern in einer Regel.

Im folgenden Beispiel einer **hosts.allow**-Datei ist es allen **example.com** Hosts gestattet, sich mit allen Diensten mit Ausnahme von **cracker.example.com** zu verbinden:

```
ALL: .example.com EXCEPT cracker.example.com
```

In einem anderen Beispiel einer **hosts.allow**-Datei können Clients des **192.168.0.*x***-Netzwerks alle Dienste benutzen, mit der Ausnahme von FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

Anmerkung

Der besseren Übersicht halber ist es oft besser, **EXCEPT**-Operatoren zu vermeiden. Dadurch können andere Administratoren schnell die gewünschten Dateien durchsuchen, um zu sehen, welche Hosts Zugriff und welche keinen Zugriff auf bestimmte Dienste haben sollen, ohne dass mehrere **EXCEPT**-Operatoren berücksichtigt werden müssen.

2.3.2.2. Optionsfelder

Zusätzlich zu den grundlegenden Regeln, die den Zugriff gewähren oder ablehnen, unterstützt die Red Hat Enterprise Linux Implementierung von TCP-Wrappern auch Erweiterungen der Zugriffskontrollsprache durch Optionsfelder. Mithilfe von Optionsfeldern innerhalb einer Hosts-Zugriffsregel können Administratoren eine Reihe von Aufgaben durchführen, wie z. B. die Änderung des Protokollierungsverhaltens, die Zusammenfassung der Zugriffskontrolle und der Ausführung von Shell-Befehlen.

2.3.2.2.1. Protokollierung

Optionsfelder ermöglichen es Administratoren, die Protokoll-Facility und die Prioritätsstufe für eine Regel einfach zu ändern, indem die **severity**-Direktive verwendet wird.

Im folgenden Beispiel werden Verbindungen zum SSH-Daemon von jedem Host in der **example.com**-Domain in die standardmäßige Protokoll-Facility **authpriv syslog** geschrieben (da kein Facility-Wert angegeben ist), und dies mit einer Priorität von **emerg**:

```
sshd : .example.com : severity emerg
```

Es ist auch möglich, eine Facility mit der **severity**-Option anzugeben. Das folgende Beispiel protokolliert alle SSH-Verbindungsversuche von Hosts aus der **example.com**-Domain zur **local0**-Facility, mit einer Priorität von **alert**:

```
sshd : .example.com : severity local0.alert
```

Anmerkung

In der Praxis wird dieses Beispiel nicht funktionieren, so lange der Syslog-Daemon (**syslogd**) nicht dazu konfiguriert ist, an die **local0**-Facility zu protokollieren. Weitere Informationen zur Konfiguration von benutzerdefinierten Facilitys finden Sie auf der **syslog.conf**-Handbuchseite.

2.3.2.2.2. Zugriffskontrolle

Optionsfelder erlauben es den Administratoren, Hosts mit einer einzelnen Regel explizit anzunehmen oder abzulehnen, indem sie die **allow** oder **deny**-Direktive als letzte Option hinzufügen.

Die folgenden Regeln etwa erlauben SSH-Verbindungen von **client-1.example.com**, lehnen aber Verbindungsversuche von **client-2.example.com** ab:

```
sshd : client-1.example.com : allow
sshd : client-2.example.com : deny
```

Indem die Zugriffskontrolle pro Regel ermöglicht wird, können Administratoren mithilfe des Optionsfelds

alle Zugriffsregeln in einer einzelnen Datei vereinen: Entweder in **hosts.allow** oder in **hosts.deny**. Einige Administratoren finden diese Art, die Zugriffsregeln zu organisieren, einfacher.

2.3.2.2.3. Shell-Befehle

Optionsfelder ermöglichen es Zugriffsregeln durch die folgenden beiden Direktiven, Shell-Befehle auszuführen:

- **spawn** — Startet einen Shell-Befehl als untergeordneten Prozess. Diese Direktive kann Aufgaben wie **/usr/sbin/safe_finger** durchführen, um weitere Informationen über den anfragenden Client zu erhalten oder spezielle Protokolldateien mit dem **echo**-Befehl erzeugen.

  Im folgenden Beispiel werden Clients, die von der **example.com**-Domain aus auf einen Telnet-Dienst zuzugreifen versuchen, unbemerkt in einer speziellen Protokolldatei aufgezeichnet:

  ```
  in.telnetd : .example.com \
   : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
   : allow
  ```

- **twist** — Ersetzt den angeforderten Dienst durch den angegebenen Befehl. Diese Direktive wird oft verwendet, um Fallen für potenzielle Eindringlinge zu stellen. Es kann auch dazu verwendet werden, um Nachrichten an verbindende Clients zu senden. Die **twist**-Direktive muss am Ende der Regelzeile stehen.

  Im folgenden Beispiel wird Clients, die von der **example.com**-Domain aus auf einen FTP-Dienst zuzugreifen versuchen, mithilfe des **echo**-Befehls eine Nachricht gesendet:

  ```
  vsftpd : .example.com \
   : twist /bin/echo "421 This domain has been black-listed. Access denied!"
  ```

Weitere Informationen zur Verwendung von Shell-Befehlsoptionen finden Sie auf der **hosts_options** Handbuchseite.

2.3.2.2.4. Erweiterungen

Erweiterungen, die zusammen mit den **spawn** und **twist**-Direktiven verwendet werden, liefern Informationen über den Client, den Server sowie die beteiligten Prozesse.

Sehen Sie nachfolgend eine Liste der unterstützten Erweiterungen:

- **%a** — Die IP-Adresse des Clients.
- **%A** — Die IP-Adresse des Servers.
- **%c** — Verschiedene Client-Informationen, wie zum Beispiel der Benutzer- und Host-Name oder der Benutzername und die IP-Adresse.
- **%d** — Der Name des Daemon-Prozesses.
- **%h** — Der Host-Name des Clients (oder IP-Adresse, wenn der Host-Name nicht verfügbar ist).
- **%H** — Der Host-Name des Servers (oder IP-Adresse, wenn der Host-Name nicht verfügbar ist).
- **%n** — Der Host-Name des Clients. Wenn dieser nicht verfügbar ist, so wird **unknown** ausgegeben. Wenn der Host-Name und die Host-Adresse des Clients nicht übereinstimmen, wird **paranoid** ausgegeben.
- **%N** — Der Host-Name des Servers. Wenn dieser nicht verfügbar ist, wird **unknown** ausgegeben. Wenn der Host-Name und die Host-Adresse des Servers nicht übereinstimmen, wird **paranoid** ausgegeben.
- **%p** — Die ID des Daemon-Prozesses.

- **%s** — Verschiedene Server-Informationen, wie zum Beispiel der Daemon-Prozess und die Host- oder IP-Adresse des Servers.
- **%u** — Der Benutzername des Clients. Wenn dieser nicht verfügbar ist, wird **unknown** ausgegeben.

Die folgende Beispielregel verwendet eine Erweiterung in Verbindung mit dem **spawn**-Befehl, um den Client-Host in einer benutzerdefinierten Protokolldatei zu identifizieren.

Sollte ein Verbindungsversuch zum SSH-Daemon (**sshd**) von einem Host in der **example.com**-Domain unternommen werden, führen Sie den **echo**-Befehl aus, um den Versuch in eine spezielle Protokolldatei zu schreiben, einschließlich des Host-Namens des Clients (unter Verwendung der **%h**-Erweiterung):

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
 : deny
```

Auf ähnliche Weise können Erweiterungen dazu verwendet werden, um Nachrichten für bestimmte Clients zu personalisieren. Im folgenden Beispiel wird Clients, die auf FTP-Dienste von der **example.com**-Domain aus zuzugreifen versuchen, mitgeteilt, dass diese vom Server ausgeschlossen wurden:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Eine vollständige Erklärung der verfügbaren Erweiterungen, sowie der zusätzlichen Zugriffskontrolloptionen finden Sie in Abschnitt 5 der Handbuchseiten von **hosts_access** (**man 5 hosts_access**) sowie der Handbuchseite für **hosts_options**.

Weitere Informationen zu TCP-Wrappern finden Sie unter Abschnitt 2.3.5, „Zusätzliche Informationsquellen“.

2.3.3. xinetd

Der **xinetd**-Daemon ist ein TCP-wrapped *Super-Dienst*, der den Zugriff auf eine Reihe gängiger Netzwerkdienste wie FTP, IMAP und Telnet steuert. Er bietet außerdem dienstspezifische Konfigurationsoptionen zur Zugriffskontrolle, erweiterte Protokollierung, Binding, Umleitungen sowie Ressourcenverwaltung.

Wenn ein Client einen Verbindungsversuch mit einem durch **xinetd** gesteuerten Netzwerkdienst unternimmt, so erhält der Super-Dienst die Anfrage und prüft auf Zugriffskontrollregeln der TCP-Wrapper.

Falls der Zugriff gestattet ist, so verifiziert **xinetd**, dass die Verbindung unter den eigenen Zugriffsregeln für diesen Dienst gestattet ist. Es wird auch geprüft, ob dem Dienst mehr Ressourcen zugewiesen werden können und dass keine definierten Regeln verletzt werden.

Falls alle diese Bedingungen erfüllt sind (d. h. der Zugriff auf den Dienst wurde gewährt; der Dienst hat seine Ressourcengrenze nicht erreicht und es werden keine definierten Regeln verletzt), so startet **xinetd** eine Instanz des angefragten Dienstes und gibt die Kontrolle über die Verbindung daran ab. Sobald die Verbindung besteht, greift **xinetd** nicht weiter in die Kommunikation zwischen Client Host und Server ein.

2.3.4. xinetd-Konfigurationsdateien

Die Konfigurationsdateien für **xinetd** lauten wie folgt:

- **`/etc/xinetd.conf`** — Die allgemeine **`xinetd`**-Konfigurationsdatei.
- **`/etc/xinetd.d/`** — Das Verzeichnis, das alle dienstspezifischen Dateien enthält.

2.3.4.1. Die /etc/xinetd.conf-Datei

Die **`/etc/xinetd.conf`**-Datei enthält allgemeine Konfigurationseinstellungen, die sich auf jeden Dienst unter der Kontrolle von **`xinetd`** auswirken. Bei jedem Start des **`xinetd`**-Dienstes wird diese Datei gelesen. Damit Konfigurationsänderungen wirksam werden, muss der Administrator den **`xinetd`**-Dienst also neu starten. Nachfolgend sehen Sie ein Beispiel für eine **`/etc/xinetd.conf`**-Datei:

```
defaults
{
    instances                  = 60
    log_type                    = SYSLOG authpriv
    log_on_success             = HOST PID
    log_on_failure              = HOST
    cps                         = 25 30
}
includedir /etc/xinetd.d
```

Diese Zeilen kontrollieren die folgenden Aspekte von **`xinetd`**:

- **`instances`** — Legt die Höchstzahl von Anfragen fest, die **`xinetd`** gleichzeitig bearbeiten kann.
- **`log_type`** — Weist **`xinetd`** an, die **`authpriv`**-Facillity zu verwenden, die Protokolleinträge in die **`/var/log/secure`**-Datei schreibt. Das Hinzufügen einer Direktive wie **`FILE /var/log/xinetdlog`** würde eine benutzerdefinierte Protokolldatei mit dem Namen **`xinetdlog`** im **`/var/log/`**-Verzeichnis erstellen.
- **`log_on_success`** — Weist **`xinetd`** dazu an, erfolgreiche Verbindungsversuche zu protokollieren. Standardmäßig werden die Remote-Host-IP-Adresse und die ID des Servers, der die Anfrage verarbeitet, aufgezeichnet.
- **`log_on_failure`** — Weist **`xinetd`** dazu an, fehlgeschlagene oder abgewiesene Verbindungsversuche zu protokollieren.
- **`cps`** — Weist **`xinetd`** dazu an, für einen bestimmten Dienst nicht mehr als 25 Verbindungen pro Sekunde zuzulassen. Wenn diese Grenze erreicht ist, wird der Dienst für 30 Sekunden ausgesetzt.
- **`includedir /etc/xinetd.d/`** — Enthält Optionen der dienstspezifischen Konfigurationsdateien im Verzeichnis **`/etc/xinetd.d/`**. Weitere Informationen zu diesem Verzeichnis finden Sie unter Abschnitt 2.3.4.2, „Das /etc/xinetd.d/-Verzeichnis“.

Anmerkung

Die Einstellungen **`log_on_success`** und **`log_on_failure`** in **`/etc/xinetd.conf`** werden oftmals von den dienstspezifischen Protokolldateien geändert. Aus diesem Grund können mehr Informationen in der Protokolldatei eines Dienstes angezeigt werden, als die **`/etc/xinetd.conf`**-Datei angibt. Weitere Informationen diesbezüglich finden Sie unter Abschnitt 2.3.4.3.1, „Protokolloptionen“.

2.3.4.2. Das /etc/xinetd.d/-Verzeichnis

Das **`/etc/xinetd.d/`**-Verzeichnis enthält die Konfigurationsdateien für jeden einzelnen Dienst, der von **`xinetd`** verwaltet wird sowie die Namen der Dateien, die mit dem Dienst zusammenhängen. Wie auch **`xinetd.conf`** wird diese Datei nur gelesen, wenn der **`xinetd`**-Dienst gestartet wird. Um

Änderungen wirksam werden zu lassen, muss der Administrator den **xinetd**-Dienst daher neu starten.

Die Dateien im **/etc/xinetd.d/**-Verzeichnis verwenden dieselben Konventionen und Optionen wie **/etc/xinetd.conf**. Der Hauptgrund dafür, dass sich diese in eigenen Konfigurationsdateien befinden, ist zur einfacheren Anpassung und um Auswirkungen auf andere Dienste möglichst zu vermeiden.

Um einen Überblick über die Struktur dieser Dateien zu erhalten, werfen Sie einen Blick auf die Datei **/etc/xinetd.d/krb5-telnet**:

```
service telnet
{
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  disable         = yes
}
```

Diese Zeilen kontrollieren die folgenden Aspekte des **telnet**-Dienstes:

- **service** — Definiert den Dienstnamen, meist einer der in der **/etc/services**-Datei aufgeführten.
- **flags** — Legt eine beliebiege Anzahl von Parametern für die Verbindung fest. **REUSE** weist **xinetd** an, den Socket für eine Telnet-Verbindung wiederzuverwenden.

Anmerkung

Das **REUSE**-Flag ist veraltet. Alle Dienste verwenden jetzt implizit das **REUSE**-Flag.

- **socket_type** — Setzt den Netzwerk-Sockettyp auf **stream**.
- **wait** — Legt fest, ob der Dienst "einthreadig" (**yes**) oder "mehrthreadig" (**no**) ist.
- **user** — Legt fest, unter welcher Benutzer-ID der Prozess abläuft.
- **server** — Legt die auszuführende Binärdatei fest.
- **log_on_failure** — Bestimmt die Protokollparameter für **log_on_failure** zusätzlich zu den in **xinetd.conf** bereits definierten.
- **disable** — Legt fest, ob der Dienst deaktiviert (**yes**) oder aktiviert (**no**) ist.

Weitere Informationen zu diesen Optionen und deren Gebrauch finden Sie auf der **xinetd.conf** Handbuchseite.

2.3.4.3. Änderungen an xinetd-Konfigurationsdateien

Es gibt eine große Anzahl an Direktiven für durch **xinetd** geschützte Dienste. Dieser Abschnitt beschreibt einige der häufig verwendeten Optionen.

2.3.4.3.1. Protokolloptionen

Die folgenden Protokolloptionen stehen für **/etc/xinetd.conf** und die dienstspezifischen Konfigurationsdateien im **/etc/xinetd.d/**-Verzeichnis zur Verfügung.

Nachfolgend sehen Sie eine Liste der häufig verwendeten Protokolloptionen:

- **ATTEMPT** — Protokolliert einen fehlgeschlagenen Versuch (**log_on_failure**).
- **DURATION** — Protokolliert, wie lange ein Remote-System einen Dienst nutzt (**log_on_success**).
- **EXIT** — protokolliert den Exit-Status oder das Endsignal des Dienstes (**log_on_success**).
- **HOST** — Protokolliert die IP-Adresse des Remote-Host-Rechners (**log_on_failure** und **log_on_success**).
- **PID** — Protokolliert die Prozess-ID des Servers, an den die Anfrage gesendet wird (**log_on_success**).
- **USERID** — Protokolliert den Remote-Benutzer mithilfe der in RFC 1413 definierten Methode für alle mehrthreadigen Stream-Dienste (**log_on_failure** und **log_on_success**).

Eine vollständige Liste der Protokolloptionen finden Sie auf der **xinetd.conf** Handbuchseite.

2.3.4.3.2. Zugriffskontroll-Optionen

Benutzer von **xinetd**-Diensten können wählen, ob sie die Host-Zugriffskontrolldateien der TCP-Wrapper, Zugriffskontrolle mittels der **xinetd**-Konfigurationsdateien oder eine Kombination aus beidem verwenden wollen. Informationen zum Gebrauch von Host-Zugriffskontrolldateien der TCP-Wrapper finden Sie in Abschnitt 2.3.2, „TCP-Wrapper Konfigurationsdateien“.

In diesem Abschnitt wird der Einsatz von **xinetd** für die Kontrolle von Zugriffen auf bestimmte Dienste behandelt.

Anmerkung

Im Gegensatz zu TCP-Wrappern muss der **xinetd**-Administrator nach jeder Änderung den **xinetd**-Dienst neu starten, damit diese wirksam werden.
Ebenfalls im Gegensatz zu TCP-Wrappern betrifft die Zugriffskontrolle durch **xinetd** lediglich die Dienste, die durch **xinetd** kontrolliert werden.

Die **xinetd**-Host-Zugriffskontrolle unterscheidet sich von der von TCP-Wrappern verwendeten Methode. Während TCP-Wrapper die gesamte Zugriffskonfiguration in zwei Dateien ablegt, **/etc/hosts.allow** und **/etc/hosts.deny**, befindet sich die **xinetd**-Zugriffskontrolle in den jeweiligen Dienstkonfigurationsdateien im **/etc/xinetd.d/**-Verzeichnis.

Die folgenden Optionen werden von **xinetd** für die Host-Zugriffskontrolle unterstützt:

- **only_from** — Erlaubt nur den aufgeführten Host-Rechnern den Zugriff auf den Dienst.
- **no_access** — Verwehrt den aufgeführten Host-Rechnern den Zugriff auf den Dienst.
- **access_times** — Der Zeitraum, in dem ein bestimmter Dienst verwendet werden darf. Der Zeitraum muss im 24-Stunden-Format, also HH:MM-HH:MM, angegeben werden.

Die Optionen **only_from** und **no_access** können eine Liste von IP-Adressen oder Hostnamen verwenden, oder ein gesamtes Netzwerk referenzieren. Wie TCP-Wrapper kann durch die Kombination der **xinetd**-Zugriffskontrolle und der entsprechenden Protokollkonfiguration die Sicherheit durch das Abweisen von Anfragen von gesperrten Hosts und das Protokollieren aller Verbindungsversuche erhöht werden.

Zum Beispiel kann die folgende **/etc/xinetd.d/telnet**-Datei verwendet werden, um den Telnet-Zugriff von einer bestimmten Netzwerkgruppe auf ein System zu verweigern und um die Zeitspanne, die selbst erlaubte Benutzer angemeldet sein dürfen, einzuschränken:

```
service telnet
{
  disable          = no
  flags            = REUSE
  socket_type      = stream
  wait             = no
  user             = root
  server           = /usr/kerberos/sbin/telnetd
  log_on_failure   += USERID
  no_access        = 172.16.45.0/24
  log_on_success   += PID HOST EXIT
  access_times     = 09:45-16:15
}
```

Wenn nun ein Client-System vom **172.16.45.0/24**-Netzwerk wie etwa von **172.16.45.2** versucht, auf den Telnet-Dienst zuzugreifen, erhält es die folgende Meldung:

```
Connection closed by foreign host.
```

Außerdem werden diese Anmeldeversuche in **/var/log/messages** protokolliert:

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285 from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285
duration=0(sec)
```

Wenn Sie TCP-Wrapper zusammen mit der Zugriffskontrolle von **xinetd** verwenden, müssen Sie die Beziehung dieser beiden Zugriffskontroll-Mechanismen zueinander verstehen.

Im Folgenden wird die Abfolge der **xinetd**-Vorgänge beschrieben, wenn ein Client eine Verbindung anfordert:

1. Der **xinetd**-Daemon greift auf die Host-Zugriffsregeln der TCP-Wrapper durch einen **libwrap.a**-Bibliotheksaufruf zu. Besteht eine Deny-Regel für den Client, so wird die Verbindung nicht aufgebaut. Besteht eine Allow-Regel für den Client, wird die Verbindung an **xinetd** weitergegeben.
2. Der **xinetd**-Daemon überprüft seine eigenen Zugriffskontrollregeln für den **xinetd**-Dienst und den angeforderten Dienst. Besteht eine Deny-Regel für den Client, wird die Verbindung nicht aufgebaut. Andernfalls startet **xinetd** eine Instanz des angeforderten Dienstes und gibt die Kontrolle über die Verbindung an diesen weiter.

Wichtig

Seien Sie vorsichtig bei der Verwendung von TCP-Wrapper-Zugriffskontrollen in Verbindung mit **xinetd**-Zugriffskontrollen. Eine Fehlkonfiguration kann unerwünschte Auswirkungen haben.

2.3.4.3.3. Bindungs- und Umleitungsoptionen

Die Dienstkonfigurationsdateien für **xinetd** unterstützen auch die Bindung des Dienstes an eine bestimmte IP-Adresse und Umleitung der eingehenden Anfragen für diesen Dienst an andere IP-Adressen, Hostnamen oder Ports.

Die Bindung wird von der **bind**-Option in den Dienstkonfigurationsdateien gesteuert und verknüpft den

Dienst mit einer IP-Adresse auf dem System. Nach der Konfiguration lässt die **bind**-Option nur Anfragen für die richtige IP-Adresse zum Zugriff auf den Dienst zu. Auf diese Weise kann jeder Dienst je nach Bedarf an verschiedene Netzwerkschnittstellen gebunden werden.

Dies ist besonders nützlich bei Systemen mit mehreren Netzwerkadaptern oder mehreren IP-Adressen. Bei solchen Systemen können unsichere Dienste (z. B. Telnet) dazu konfiguriert werden, nur auf die Schnittstelle zu horchen, die mit einem privaten Netzwerk verbunden ist, nicht auf die Schnittstelle zum Internet.

Die **redirect**-Option akzeptiert eine IP-Adresse oder einen Hostnamen gefolgt von einer Portnummer. Sie konfiguriert den Dienst, um alle Anfragen für diesen Dienst an eine bestimmte Adresse und Portnummer weiterzuleiten. Diese Option kann verwendet werden, um auf eine andere Portnummer auf demselben System zu verweisen, die Anfrage an eine andere IP-Adresse auf demselben Rechner weiterzuleiten, die Anfrage an ein anderes System oder eine andere Portnummer zu verschieben, oder aber eine Kombination all dieser Optionen. Auf diese Weise kann ein Benutzer, der sich für einen bestimmten Dienst an einem System anmeldet, ohne Unterbrechung umgeleitet werden.

Der **xinetd**-Daemon kann diese Umleitung durch Erzeugen eines Prozesses ausführen, der während der Verbindung des anfragenden Client-Rechners mit dem Host-Rechner, der den eigentlichen Dienst liefert, im Stay-Alive-Modus läuft und Daten zwischen den zwei Systemen austauscht.

Die eigentliche Stärke der **bind** und **redirect**-Optionen liegt in deren kombinierter Verwendung. Durch Bindung eines Dienstes an eine bestimmte IP-Adresse auf einem System und dem darauffolgenden Umleiten der Anfragen für denselben Dienst an einen zweiten Rechner, der nur für den ersten Rechner sichtbar ist, können Sie ein internes System verwenden, um Dienste für vollkommen unterschiedliche Netzwerke zur Verfügung zu stellen. Alternativ können diese Optionen verwendet werden, damit ein Dienst auf einem Multihomed-Rechner weniger einer bekannten IP-Adresse ausgesetzt ist, und um jegliche Anfragen für diesen Dienst an einen anderen Rechner weiterzuleiten, der eigens für diesen Zweck konfiguriert ist.

Nehmen wir zum Beispiel ein System, das mit diesen Einstellungen für seinen Telnet-Dienst als Firewall verwendet wird:

```
service telnet
{
  socket_type  = stream
  wait   = no
  server   = /usr/kerberos/sbin/telnetd
  log_on_success  += DURATION USERID
  log_on_failure  += USERID
  bind                    = 123.123.123.123
  redirect                = 10.0.1.13 23
}
```

Die **bind** und **redirect**-Optionen in dieser Datei stellen sicher, dass der Telnet-Dienst auf dem Rechner an eine externe IP-Adresse (**123.123.123.123**) gebunden ist, und zwar die Internet-seitige. Außerdem werden alle an **123.123.123.123** gesendeten Telnet-Anfragen über einen zweiten Netzwerkadapter an eine interne IP-Adresse (**10.0.1.13**) weitergeleitet, auf die nur die Firewall und interne Systeme Zugriff haben. Die Firewall sendet dann die Kommunikation von einem System an das andere, und für das sich verbindende System sieht es so aus, als ob es mit **123.123.123.123** verbunden sei, während es in Wirklichkeit mit einem anderen Rechner verbunden ist.

Diese Funktion ist besonders nützlich für Benutzer mit Breitbandverbindungen und nur einer festen IP-Adresse. Wird Network Address Translation (NAT) verwendet, sind die Systeme hinter dem Gateway-Rechner, die nur interne IP-Adressen verwenden, außerhalb des Gateway-Systems nicht verfügbar.

Wenn jedoch bestimmte Dienste, die von **xinetd** kontrolliert werden, mit den Optionen **bind** und **redirect** konfiguriert sind, kann der Gateway-Rechner als eine Art Proxy zwischen externen Systemen und einem bestimmten internen Rechner fungieren, der konfiguriert ist, um den Dienst breitzustellen. Außerdem sind die verschiedenen **xinetd**-Zugriffskontroll- und Protokollierungsoptionen auch für zusätzlichen Schutz verfügbar.

2.3.4.3.4. Optionen zur Ressourcenverwaltung

Der **xinetd**-Daemon kann einen einfachen Grad an Schutz vor Denial of Service (DoS) Angriffen bieten. Untenstehend finden Sie eine Liste an Direktiven, welche die Auswirkung dieser Angriffe abschwächen können:

- **per_source** — Legt die Höchstanzahl von Verbindungen von einer bestimmen IP-Adresse mit einem bestimmen Dienst fest. Es werden nur ganzzahlige Werte als Parameter akzeptiert. Diese Direktive kann sowohl in **xinetd.conf** als auch in den dienstspezifischen Konfigurationsdateien im **xinetd.d/**-Verzeichnis verwendet werden.
- **cps** — Legt die Höchstzahl der Verbindungen pro Sekunde fest. Diese Option akzeptiert zwei ganzzahlige Parameter getrennt durch eine Leerstelle. Die erste Zahl ist die Höchstzahl von Verbindungen zum Dienst pro Sekunde. Die zweite Zahl ist die Anzahl der Sekunden, die **xinetd** warten muss, bis der Dienst erneut aktiviert wird. Es werden nur ganzzahlige Werte als Parameter akzeptiert. Diese Direktive kann sowohl in **xinetd.conf** als auch in den dienstspezifischen Konfigurationsdateien im **xinetd.d/**-Verzeichnis verwendet werden.
- **max_load** — Legt den Schwellenwert für die CPU-Nutzung oder durchschnittliche Auslastung eines Dienstes fest. Es akzeptiert deine Gleitkommazahl als Parameter.

  Die durchschnittliche Auslastung ist ein ungefähres Maß dafür, wie viele Prozesse zu einem bestimmten Zeitpunkt aktiv sind. Weitere Informationen zur durchschnittlichen Auslastung finden Sie unter den **uptime**, **who** und **procinfo**-Befehlen.

Es gibt noch weitere Optionen für die Ressourcenverwaltung mit **xinetd**. Auf der **xinetd.conf** Handbuchseite finden Sie weitere Informationen diesbezüglich.

2.3.5. Zusätzliche Informationsquellen

Weitere Informationen zu TCP-Wrappern und **xinetd** finden Sie in der Systemdokumentation und im Internet.

2.3.5.1. Installierte TCP-Wrapper-Dokumentation

Die Dokumentation auf Ihrem System ist ein guter Ausgangspunkt, wenn Sie weitere Informationen zu TCP-Wrappern, **xinetd** und zur Zugriffskontrolle suchen.

- **/usr/share/doc/tcp_wrappers-<*version*>/** — Dieses Verzeichnis enthält eine **README**-Datei, in der die Funktionsweise von TCP-Wrappern und die verschiedenen Hostname- und Hostadress-Spoofing-Risiken beschrieben werden.
- **/usr/share/doc/xinetd-<*version*>/** — Dieses Verzeichnis enthält eine **README**-Datei, in der Aspekte der Zugriffskontrolle beschrieben sind und eine **sample.conf**-Datei mit verschiedenen Ideen zum Bearbeiten der Konfigurationsdateien im **/etc/xinetd.d/**-Verzeichnis.
- Handbuchseiten zu TCP-Wrappern und **xinetd** — Es gibt eine Reihe von Handbuchseiten für die verschiedenen Applikationen und Konfigurationsdateien rund um TCP-Wrapper und **xinetd**. Die folgende Liste benennt einige der wichtigeren Handbuchseiten.

  Server-Applikationen

  - **man xinetd** — Die Handbuchseite für **xinetd**.

Konfigurationsdateien

- **`man 5 hosts_access`** — Die Handbuchseite für die Hosts-Zugriffskontrolldateien der TCP-Wrapper.
- **`man hosts_options`** — Die Handbuchseite für die Optionsfelder der TCP-Wrapper.
- **`man xinetd.conf`** — Die Handbuchseite mit einer Liste der **`xinetd`**-Konfigurationsoptionen.

2.3.5.2. Hilfreiche TCP-Wrapper-Websites

- http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial — Eine ausführliche Anleitung, in der viele Möglichkeiten beschrieben werden, standardmäßige **`xinetd`**-Konfigurationsdateien für bestimmte Sicherheitsanforderungen anzupassen.

2.3.5.3. Bücher zum Thema

- *Hacking Linux Exposed* von Brian Hatch, James Lee und George Kurtz; Osbourne/McGraw-Hill — Eine exzellente Informationsquelle zu TCP-Wrappern und **`xinetd`**.

2.4. Virtual Private Networks (VPNs)

Unternehmen mit mehreren Zweigstellen sind häufig über spezielle Leitungen miteinander verbunden, um die Effizienz und den Schutz sensibler Daten zu gewährleisten. Viele Unternehmen nutzen zum Beispiel Frame Relay oder *Asynchronous Transfer Mode* (ATM) Leitungen als Netzwerklösung, um Büros miteinander zu verbinden. Dies kann jedoch eine teure Lösung sein, insbesondere für kleine bis mittelständische Unternehmen, die sich zwar vergrößern möchten, jedoch nicht die hohen Kosten für dedizierte Digitalleitungen der Unternehmensklasse in Kauf nehmen wollen.

Virtual Private Networks (kurz VPN) stellen eine Lösung für dieses Problem dar. Dem Prinzip dedizierter Digitalschaltungen folgend, ermöglichen VPNs gesicherte, digitale Kommunikation zwischen zwei Parteien (oder Netzwerken) und bilden somit ein *Wide-Area-Netzwerk* (WAN) aus bestehenden *Local-Area-Netzwerken* (LANs). Der Unterschied zum Frame Relay oder ATM ist das Transportmedium. VPNs übertragen via IP-Datagrammen, und sorgen somit für eine sichere Übertragung über das Internet zum Bestimmungsort. Die meisten frei verfügbaren VPN-Implementierungen verwenden offene Standards als Verschlüsselungsmethode, um die Daten während der Übertragung weiter zu maskieren.

Einige Unternehmen setzen VPN-Hardware-Lösungen ein, um die Sicherheit zu erhöhen, während andere Software- oder Protokoll-basierte Implementierungen verwenden. Es gibt mehrere Hersteller für Hardware-VPN-Lösungen, wie z. B. Cisco, Nortel, IBM und Checkpoint. Es gibt eine kostenlose, Software-basierte VPN-Lösung für Linux namens FreeS/Wan, die eine standardisierte IPSec (*Internet Protocol Security*) Implementierung verwendet. Diese VPN-Lösungen, egal ob Hardware- oder Software-basiert, verhalten sich wie spezielle Router, die sich zwischen der IP-Verbindung von einem Büro zum anderen befinden.

2.4.1. Funktionsweise eines VPNs

Wenn ein Paket von einem Client verschickt wird, wird es durch den VPN-Router oder -Gateway gesendet. Dieser fügt ein *Authentication Header* (AH) für das Routing und zur Authentifizierung hinzu. Die Daten werden dann verschlüsselt und schließlich in ein *Encapsulating Security Payload* (ESP) Paket eingeschlossen. Letzteres enthält die Verschlüsselung und Anweisungen zur Verarbeitung.

Der empfangende VPN-Router holt sich die Header-Information, entschlüsselt die Daten und leitet diese zum Zielort weiter (entweder an einen Arbeitsplatzrechner oder einen Knoten im Netzwerk). Unter Verwendung einer Netzwerk-zu-Netzwerk Verbindung erhält der empfangende Knoten am lokalen

Netzwerk die Pakete schon entschlüsselt und bereit zur Verarbeitung. Der Verschlüsselungs- /Entschlüsselungsprozess in einer Netzwerk-zu-Netzwerk VPN-Verbindung ist für den lokalen Knoten transparent.

Durch solch einen erhöhten Grad an Sicherheit muss ein Angreifer nicht nur ein Paket abfangen, sondern dies auch noch entschlüsseln. Angreifer, die eine Man-in-the-Middle-Attacke zwischen einem Server und einem Client durchführen, müssen daher auch Zugang zu mindestens einem der privaten Schlüssel besitzen, die für die Authentifizierung der Sessions verwendet werden. Aufgrund ihrer verschiedenen Schichten zur Authentifizierung und Verschlüsselung sind VPNs ein sicheres und effektives Mittel für die Verbindung mehrerer entfernter Knoten, die sich dadurch wie ein einziges Intranet verhalten können.

2.4.2. Openswan

2.4.2.1. Überblick

Überblick

Openswan ist eine quelloffene IPsec-Implementierung auf Kernel-Ebene in Red Hat Enterprise Linux. Es setzt die IKE-Protokolle (Internet Key Exchange) v1 und v2 zur Schlüsselverwaltung ein, implementiert als Daemons auf Benutzerebene. Manuelle Einrichtung von Schlüsseln ist ebenfalls möglich mittels der **`ip xfrm`** Befehle, dies wird jedoch nicht empfohlen.

Kryptografische Unterstützung

Openswan verfügt über eine integrierte kryptografische Bibliothek, unterstützt jedoch auch eine NSS (Network Security Services) Bibliothek, die vollständig unterstützt wird und zur Einhaltung der FIPS-Sicherheitsstandards notwendig ist. Weitere Informationen über FIPS (Federal Information Processing Standard) finden Sie unter Abschnitt 7.2, „Federal Information Processing Standard (FIPS)“.

Installation

Führen Sie den Befehl **`yum install openswan`** aus, um Openswan zu installieren.

2.4.2.2. Konfiguration

Speicherorte

Dieser Abschnitt erläutert die Dateien und Verzeichnisse, die zur Konfiguration von Openswan wichtig sind.

- **`/etc/ipsec.d`** - Hauptverzeichnis. Speichert die Dateien im Zusammenhang mit Openswan.
- **`/etc/ipsec.conf`** - Hauptkonfigurationsdatei. Weitere **`*.conf`**-Konfigurationsdateien für individuelle Konfigurationen können in **`/etc/ipsec.d`** erstellt werden.
- **`/etc/ipsec.secrets`** - Hauptgeheimnisdatei. Weitere **`*.secrets`**-Dateien für individuelle Konfigurationen können in **`/etc/ipsec.d`** erstellt werden.
- **`/etc/ipsec.d/cert*.db`** - Zertifikatsdatenbankdateien. Die alte standardmäßige NSS-Datenbankdatei ist **`cert8.db`**. Ab Red Hat Enterprise Linux 6 werden NSS SQLite-Datenbanken in der **`cert9.db`**-Datei verwendet.
- **`/etc/ipsec.d/key*.db`** - Schlüssedatenbankdateien. Die alte standardmäßige NSS-Datenbankdatei ist **`key3.db`**. Ab Red Hat Enterprise Linux 6 werden NSS SQLite-Datenbanken in der **`cert9.db`**-Datei verwendet.
- **`/etc/ipsec.d/cacerts`** - Speicherort für Zertifikate von Zertifikatsstellen (auch Certificate

Authorities, kurz CA).

- **/etc/ipsec.d/certs** - Speicherort für Benutzerzertifikate. Nicht notwendig bei der Verwendung von NSS.
- **/etc/ipsec.d/policies** - Gruppenrichtlinien. Richtlinien können als *block*, *clear*, *clear-or-private*, *private* oder *private-or-clear* definiert werden.
- **/etc/ipsec.d/nsspassword** - NSS-Passwortdatei. Diese Datei ist standardmäßig nicht vorhanden, wird jedoch benötigt, falls die NSS-Datenbank mit einem Passwort erstellt wird.

Konfigurationsparameter

Dieser Abschnitt listet einige der verfügbaren Konfigurationsoptionen auf, von denen die meisten in **/etc/ipsec.conf** gespeichert werden.

- **protostack** - definert den verwendeten Protokollstapel. Die standardmäßige Option in Red Hat Enterprise Linux 6 ist *netkey*. Andere gültige Werte sind *auto*, *klips* und *mast*.
- **nat_traversal** - definiert, ob NAT für Verbindungen akzeptiert wird. Standardmäßig ist dies nicht der Fall.
- **dumpdir** - definiert den Speicherort für Speicherauszugsdateien.
- **nhelpers** - falls NSS eingesetzt wird, definiert dies die Anzahl der Threads, die für kryptografische Operationen verwendet werden. Falls NSS nicht eingesetzt wird, definiert dies die Anzahl der Prozesse, die für kryptografische Operationen verwendet werden.
- **virtual_private** - erlaubte Subnetze für die Client-Verbindung. Bereiche, die sich hinter einem NAT-Router befinden können, über den ein Client verbindet.
- **plutorestartoncrash** - standardmäßig auf "yes" gesetzt.
- **plutostderr** - Pfad zum Puto-Fehlerprotokoll. Verweist standardmäßig auf den syslog-Speicherort.
- **connaddrfamily** - kann entweder auf ipv4 oder ipv6 gesetzt werden.

Weitere Details über die Openswan-Konfiguration finden Sie auf der **ipsec.conf(5)**-Handbuchseite.

2.4.2.3. Befehle

Dieser Abschnitt erläutert die für Openswan verwendeten Befehle und zeigt entsprechende Beispiele.

Anmerkung

Wie im folgenden Beispiel gezeigt, wird die Verwendung von **service ipsec start/stop** empfohlen, um den Status des ipsec-Dienstes zu ändern. Dies ist auch die empfohlene Methode zum Starten und Stoppen aller anderen Dienste in Red Hat Enterprise Linux 6.

- Starten und Stoppen von Openswan:
  - **ipsec setup start/stop**
  - **service ipsec start/stop**
- Hinzufügen und Löschen einer Verbindung:
  - **ipsec auto --add/delete <connection name>**
- Erstellen und Abbrechen einer Verbindung
  - **ipsec auto --up/down <connection-name>**
- Generieren von RSA-Schlüsseln:
  - **ipsec newhostkey --configdir /etc/ipsec.d --password password --output**

/etc/ipsec.d/<name-of-file>

- Überprüfen von ipsec-Richtlinien im Kernel:
  - **ip xfrm policy**
  - **ip xfrm state**
- Erstellen eines selbst signierten Zertifikats:
  - **certutil -S -k rsa -n <ca-cert-nickname> -s "CN=ca-cert-common-name" -w 12 -t "C,C,C" -x -d /etc/ipsec.d**
- Erstellen von Benutzerzertifikaten signiert durch die vorherige CA:
  - **certutil -S -k rsa -c <ca-cert-nickname> -n <user-cert-nickname> -s "CN=user-cert-common-name" -w 12 -t "u,u,u" -d /etc/ipsec.d**

2.4.2.4. Informationsquellen zu Openswan

- http://www.openswan.org
- http://lists.openswan.org/pipermail/users/
- http://lists.openswan.org/pipermail/dev/
- http://www.mozilla.org/projects/security/pki/nss/
- Das *Openswan-doc*-Paket: HTML, Beispiele, README.*
- README.nss

2.5. Firewalls

Die Informationssicherheit wird üblicherweise als Prozess und nicht als Produkt angesehen. Allerdings setzen standardmäßige Sicherheitsimplementierungen normalerweise gewisse Mechanismen ein, um Zugriffsrechte zu steuern und Netzwerkressourcen auf Benutzer zu beschränken, die autorisiert, identifizierbar und nachverfolgbar sind. Red Hat Enterprise Linux enthält mehrere Tools, die Administratoren und Sicherheitstechnikern bei Fragen der Zugangskontrolle auf Netzwerkebene unterstützen können.

Firewalls sind einer der Kernbestandteile bei einer Sicherheitsimplementierung im Netzwerk. Mehrere Hersteller bieten Firewall-Lösungen an, die für alle Bereiche des Marktes geeignet sind: vom privaten Benutzer zum Schutz eines einzelnen PCs, bis hin zu Lösungen für Datenzentren, wo wichtige Unternehmensinformationen geschützt werden. Firewalls können eigenständige Hardware-Lösungen sein, wie die Firewall-Geräte von Cisco, Nokia und Sonicwall. Anbieter wie z. B. Checkpoint, McAfee und Symantec haben zudem proprietäre Software-Firewall-Lösungen für den Heim-und Firmenbereich entwickelt.

Neben den Unterschieden zwischen Hardware- und Software-Firewalls gibt es auch Unterschiede in der Funktionsweise von Firewalls, die die verschiedenen Lösungen voneinander abheben. Tabelle 2.2, „Firewall-Typen“ erklärt drei gängige Firewall-Typen und deren Funktionsweise:

Tabelle 2.2. Firewall-Typen

Methode	Beschreibung	Vorteile	Nachteile
NAT	*Network Address Translation* (NAT) platziert private IP-Subnetzwerke hinter eine einzige oder eine kleine Gruppe von externen IP-Adressen, wodurch alle Anfragen wie von einer Quelle erscheinen statt von mehreren. Der Linux-Kernel hat eine integrierte NAT-Funktionalität durch das Netfilter-Kernel-Subsystem.	· Kann transparent auf Rechnern auf einem LAN konfiguriert werden · Der Schutz vieler Rechner und Dienste hinter einer oder mehreren externen IP-Adressen vereinfacht die Verwaltung · Benutzerzugriff vom bzw. auf das LAN kann konfiguriert werden, indem Ports auf der NAT Firewall/Gateway geöffnet bzw. geschlossen werden	· Kann keine bösartigen Aktivitäten verhindern, sobald sich Benutzer mit einem Dienst außerhalb der Firewall verbinden
Paketfilter	Paketfilter-Firewalls lesen alle Datenpakete, die sich im LAN bewegen. Pakete können mithilfe der Kopfzeileninformation gelesen und bearbeitet werden. Die Pakete werden auf der Grundlage von programmierbaren Regeln gefiltert, die vom Administrator der Firewall aufgestellt wurden. Der Linux-Kernel hat eine integrierte Paketfilterfunktion über das Netfilter-Kernel-Subsystem.	· Anpassbar mithilfe des **`iptables`** Frontend-Hilfsprogramms · Erfordert keinerlei Anpassungen auf Client-Seite, da sämtliche Netzwerkaktivität auf Routerebene statt auf Applikationsebene gefiltert wird · Da Pakete keinen Proxy passieren, ist die Netzwerkgeschwindigkeit aufgrund der direkten Verbindungen vom Client zum Remote-Host höher	· Kann Pakete nicht nach Inhalt filtern wie Proxy-Firewalls · Verarbeitet Pakete auf Protokollebene, kann Pakete jedoch nicht auf Applikationsebene filtern · Komplexe Netzwerkarchitekturen erschweren das Einrichten von Paketfilter-Regeln, insbesondere wenn es zusammen mit *IP-Masquerading* oder lokalen Subnetzen und DMZ-Netzwerken eingesetzt wird
Proxy	Proxy-Firewalls filtern alle Anfragen eines bestimmten Protokolls oder Typs von den LAN-Clients zu einer Proxy-Maschine, von wo aus die Anfragen im Auftrag des lokalen Clients an das Internet gestellt werden. Eine Proxy-Maschine fungiert als ein Puffer zwischen bösartigen Benutzern von außen und den internen Client- Maschinen des	· Ermöglicht Administratoren Kontrolle darüber, welche Applikationen und Protkolle außerhalb des LAN funktionieren sollen · Einige Proxy-Server können Daten, auf die häufiger zugegriffen wird, lokal zwischenspeichern, so dass diese Daten nicht jedesmal neu über die Internetverbindung abgefragt	· Proxys sind oft applikationsspezifisch (HTTP, Telnet, etc.), oder beschränkt auf ein Protokoll (die meisten Proxys funktionieren ausschließlich mit Diensten, die über TCP verbinden) · Applikationsdienste können nicht hinter einem Proxy ausgeführt werden, Ihre Applikationsserver müssen

Netzwerkes.	Internetverbindung abgefragt werden müssen. Dadurch wird Bandbreite gespart · Proxy-Dienste können genauestens überwacht und protokolliert werden, wodurch eine bessere Kontrolle des Ressourcenverbrauchs auf dem Netzwerk ermöglicht wird	Applikationsserver müssen demnach eine separate Form der Netzwerksicherheit verwenden · Proxys können zu einem Engpass im Netzwerk werden, da alle Anfragen und Übertragungen diese eine Stelle passieren müssen, statt direkt vom Client zum entfernten Dienst zu verbinden

2.5.1. Netfilter und IPTables

Der Linux-Kernel enthält ein leistungsstarkes Netzwerk-Subsystem namens *Netfilter*. Das Netfilter-Subsystem bietet eine Paketfilterung mit oder ohne Status sowie NAT- und IP-Maskierungsdienste. Netfilter ist zudem dazu in der Lage, IP-Kopfzeileninformation für fortgeschrittenes Routing und zur Überprüfung des Verbindungszustandes zu überarbeiten (engl. *mangle*). Netfilter wird durch das **`iptables`**-Hilfsprogramm gesteuert.

2.5.1.1. Überblick über IPTables

Die Leistungsstärke und Flexibilität von Netfilter wird mithilfe des **`iptables`**-Verwaltungstools implementiert, ein Befehlszeilentool, das eine ähnliche Syntax wie sein Vorgänger **`ipchains`** verwendet. **`ipchains`** wurde ab dem Linux-Kernel 2.4 durch Netfilter/iptables abgelöst.

`iptables` verwendet das Netfilter-Subsystem zur Erweiterung, Untersuchung und Verarbeitung der Netzwerkverbindungen. **`iptables`** bietet verbesserte Protokollierung, Pre- und Post-Routing Aktionen, Network Address Translation und Port-Weiterleitung, alles in einer einzigen Befehlszeilenschnittstelle.

Dieser Abschnitt enthält eine Übersicht über **`iptables`**. Für weitere Informationen werfen Sie bitte einen Blick auf Abschnitt 2.6, „IPTables“.

2.5.2. Grundlegende Firewall-Konfiguration

Vergleichbar mit einer Brandmauer in einem Gebäude, die das Ausbreiten eines Feuers verhindern soll, so soll eine Firewall das Ausbreiten schädlicher Software auf Ihrem Computer verhindern. Sie hilft außerdem dabei, unberechtigten Benutzern den Zugriff auf Ihren Computer zu verwehren.

In einer standardmäßigen Red Hat Enterprise Linux Installation befindet sich eine Firewall zwischen Ihrem Computer oder Netzwerk und allen nicht vertrauenswürdigen Netzwerken, wie z. B. dem Internet. Die Firewall legt fest, auf welche Dienste auf Ihrem Computer Benutzer von Remote aus zugreifen können. Eine ordnungsgemäß konfigurierte Firewall kann die Sicherheit Ihres Systems signifikant erhöhen. Wir empfehlen Ihnen, für jedes Red Hat Enterprise Linux System mit Internetverbindung eine Firewall einzurichten.

2.5.2.1. Firewall-Konfigurationstool

Während der Red Hat Enterprise Linux Installation hatten Sie auf dem Bildschirm zur **`Firewall-Konfiguration`** bereits die Möglichkeit, eine einfache Firewall zu aktivieren und bestimmte Geräte, eingehende Dienste und Ports festzulegen.

Nach der Installation können Sie die dort vorgenommenen Einstellungen mithilfe des **Firewall-**

Konfigurationstools weiter anpassen.

Führen Sie den folgenden Befehl aus, um diese Applikation zu starten:

```
[root@myServer ~] # system-config-firewall
```

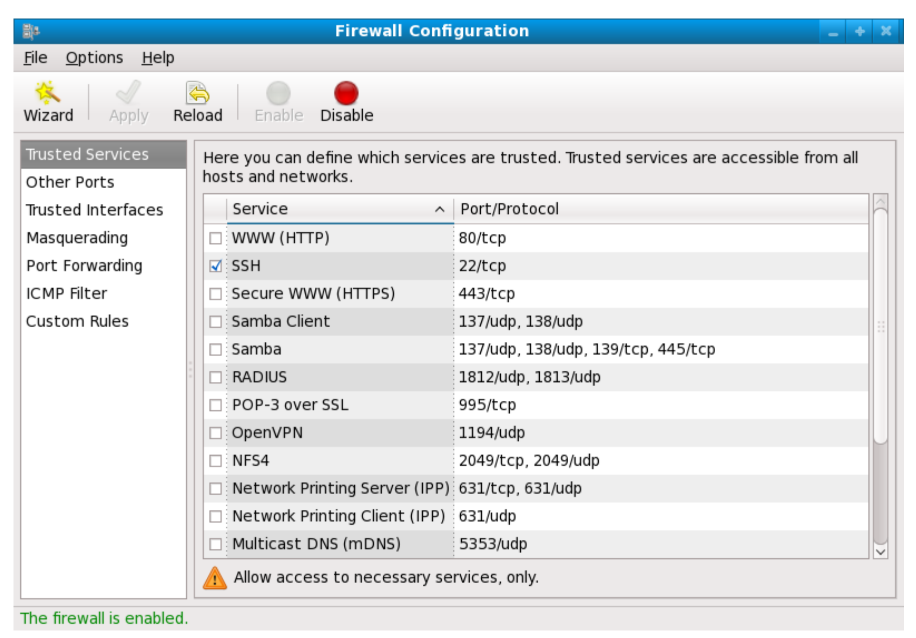

Abbildung 2.5. Firewall-Konfigurationstool

Anmerkung

Das **Firewall-Konfigurationstool** konfiguriert lediglich eine sehr einfache Firewall. Falls komplexere Regeln für das System nötig sind, werfen Sie bitte einen Blick auf Abschnitt 2.6, „IPTables“ für Informationen über die Konfiguration spezifischer **`iptables`**-Regeln.

2.5.2.2. Aktivieren und Deaktivieren der Firewall

Wählen Sie eine der folgenden Optionen für die Firewall:

- **`Deaktiviert`** — Mit deaktivierter Firewall werden keinerlei Sicherheitsprüfungen durchgeführt und der Zugang zu Ihrem System steht weit offen. Wählen Sie diese Einstellung nur, wenn sich Ihr System in einem vertrauenswürdigen Netzwerk befindet (nicht dem Internet) oder falls Sie mithilfe des iptables-Befehlszeilentools eine angepasste Firewall konfigurieren möchten.

  Warnung

  Firewall-Konfigurationen und benutzerdefinierte Firewall-Regeln werden in der **`/etc/sysconfig/iptables`**-Datei gespeichert. Falls Sie **`Deaktivieren`** wählen und auf **`OK`** klicken, gehen diese Konfigurationen und Firewall-Regeln verloren.

- **`Aktiviert`** — Diese Option konfiguriert das System derart, dass eingehende Verbindungen, die keine Antworten auf ausgehende Anfragen sind, wie z. B. DNS-Antworten oder DHCP-Anfragen, abgewiesen werden. Falls der Zugriff auf Dienste nötig ist, die auf diesem Rechner laufen, können Sie bestimmte Dienste durch die Firewall erlauben.

Falls Sie Ihr System mit dem Internet verbinden, jedoch nicht beabsichtigen, einen Server auszuführen, ist dies die sicherste Wahl.

2.5.2.3. Vertrauenswürdige Dienste

Wenn Sie Optionen in der Liste der **Vertrauenswürdige Dienste** aktivieren, wird diesen ausgewählten Diensten erlaubt, die Firewall zu passieren.

WWW (HTTP)

Das HTTP-Protokoll wird von Apache (und anderen Webservern) zur Bereitstellung von Webseiten genutzt. Falls Sie beabsichtigen, Ihren Webserver öffentlich verfügbar zu machen, markieren Sie dieses Auswahlkästchen. Diese Option ist dagegen nicht nötig, wenn Sie die Seiten nur lokal anzeigen möchten oder während Sie die Websites entwickeln. Dieser Dienst erfordert die Installation des **httpd**-Pakets.

Wenn Sie nur **WWW (HTTP)** aktivieren, wird kein Port für HTTPS geöffnet, die SSL-Version von HTTP. Falls dieser Dienst erforderlich ist, markieren Sie ebenfalls das **Secure WWW (HTTPS)** Auswahlkästchen.

FTP

Das FTP-Protokoll wird zur Übertragung von Dateien zwischen Rechnern auf einem Netzwerk verwendet. Falls Sie beabsichtigen, Ihren FTP-Server öffentlich verfügbar zu machen, markieren Sie dieses Auswahlkästchen. Dieser Dienst erfordert die Installation des **vsftpd**-Pakets.

SSH

Secure Shell (SSH) ist eine Tool-Suite zum Anmelden auf einem entfernten Rechner und zum Ausführen von Befehlen darauf. Um Fernzugriff auf den Rechner über SSH zu erlauben, markieren Sie dieses Auswahlkästchen. Dieser Dienst erfordert die Installation des **openssh-server**-Pakets.

Telnet

Telnet ist ein Protokoll zum Anmelden auf entfernen Rechnern. Die Kommunikation über Telnet ist nicht verschlüsselt und bietet keinerlei Schutz gegen das Abfangen der Daten. Es wird daher nicht empfohlen, eingehenden Telnet-Zugriff zu erlauben. Um den Fernzugriff auf den Rechner über Telnet zu erlauben, markieren Sie dieses Auswahlkästchen. Dieser Dienst erfordert die Installation des **telnet-server**-Pakets.

Mail (SMTP)

SMTP ist ein Protokoll, dass es entfernten Hosts erlaubt, zum Zustellen von E-Mail direkt mit Ihrem Rechner zu verbinden. Sie brauchen diesen Dienst nicht zu aktivieren, wenn Sie Ihre E-Mails mittels POP3 oder IMAP von dem Server Ihres Internet Service Providers abrufen, oder falls Sie ein Tool wie z. B. **fetchmail** verwenden. Um die Zustellung von E-Mail an Ihren Rechner zu erlauben, markieren Sie dieses Auswahlkästchen. Beachten Sie, dass ein fehlerhaft konfigurierter SMTP-Server es entfernten Rechnern ermöglichen kann, Ihren Server zum Versenden von Spam zu missbrauchen.

NFS4

Das Network File System (NFS) ist ein File-Sharing-Protokoll, das häufig auf *NIX-Systemen eingesetzt wird. Version 4 dieses Protokolls ist sicherer als seine Vorläufer. Falls Sie Dateien

oder Verzeichnisse auf Ihrem System für andere Benutzer auf dem Netzwerk freigeben möchten, markieren Sie dieses Auswahlkästchen.

Samba

Samba ist eine Implementierung von Microsofts proprietärem SMB-Netzwerkprotokoll. Falls Sie Dateien, Verzeichnisse oder lokal angeschlossene Drucker für Microsoft Windows Rechner auf dem Netzwerk freigeben möchten, markieren Sie dieses Auswahlkästchen.

2.5.2.4. Andere Ports

Das **Firewall-Konfigurationstool** enthält einen Abschnitt namens **Andere Ports**, um benutzerdefinierte IP-Ports für **iptables** als vertrauenswürdig festzulegen. Um beispielsweise IRC und Internet Printing Protocol (IPP) das Passieren der Firewall zu erlauben, fügen Sie Folgendes zum **Andere Ports** Abschnitt hinzu:

194:tcp,631:tcp

2.5.2.5. Speichern der Einstellungen

Klicken Sie auf **OK**, um die Änderungen zu speichern und die Firewall zu aktivieren bzw. zu deaktivieren. Falls **Firewall aktivieren** ausgewählt wurde, werden die gewählten Optionen nun in **iptables**-Befehle übersetzt und in die **/etc/sysconfig/iptables**-Datei geschrieben. Zudem wird der **iptables**-Dienst gestartet, so dass die Firewall sofort nach Abspeichern der gewählten Optionen aktiviert ist. Falls **Firewall deaktivieren** ausgewählt wurde, wird die **/etc/sysconfig/iptables**-Datei gelöscht und der **iptables**-Dienst umgehend gestoppt.

Die gewählten Optionen werden zudem in die Datei **/etc/sysconfig/system-config-firewall** geschrieben, so dass die Einstellungen beim nächsten Start der Applikation wiederhergestellt werden können. Bearbeiten Sie diese Datei nicht manuell.

Obwohl die Firewall sofort aktiviert wird, ist der **iptables**-Dienst nicht zum automatischen Start beim Systemstart konfiguriert. Siehe Abschnitt 2.5.2.6, „Aktivieren des IPTables-Dienstes“ für weitere Informationen.

2.5.2.6. Aktivieren des IPTables-Dienstes

Die Firewall-Regeln sind nur aktiv, wenn der **iptables**-Dienst läuft. Um den Dienst manuell zu starten, führen Sie den folgenden Befehl aus:

```
[root@myServer ~] # service iptables restart
```

Um zu gewährleisten, dass **iptables** zum Zeitpunkt des Systemstarts ebenfalls gestartet wird, verwenden Sie den folgenden Befehl:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

2.5.3. Verwenden von IPTables

Um **iptables** verwenden zu können, müssen Sie zunächst den **iptables**-Dienst starten. Führen Sie den folgenden Befehl aus, um den **iptables**-Dienst zu starten:

```
[root@myServer ~] # service iptables start
```

Anmerkung

Der **ip6tables**-Dienst kann ausgeschaltet werden, falls Sie ausschließlich den **iptables**-Dienst nutzen möchten. Falls Sie den **ip6tables**-Dienst deaktivieren, vergessen Sie nicht, auch das IPv6-Netzwerk zu deaktivieren. Lassen Sie nie ein Netzwerkgerät aktiv, das keine entsprechende Firewall besitzt.

Um **iptables** dazu zu zwingen, beim Hochfahren des Systems ebenfalls zu starten, führen Sie den folgenden Befehl aus:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

Dadurch wird **iptables** dazu gezwungen zu starten, sobald das System in Runlevel 3, 4 oder 5 hochgefahren wird.

2.5.3.1. Befehlssyntax von IPTables

Das folgende Beispiel eines **iptables**-Befehls veranschaulicht die grundlegende Befehlssyntax:

```
[root@myServer ~ ] # iptables -A <chain> -j <target>
```

Die **-A**-Option gibt an, dass die Regel ans Ende der *<chain>* (Kette) angehängt werden soll. Jede Kette besteht aus einer oder mehrerer *rules* (Regeln), und wird daher auch als *Regelset* bezeichnet.

Die drei integrierten Ketten sind INPUT, OUTPUT und FORWARD. Diese Ketten sind dauerhaft integriert und können nicht gelöscht werden. Die Kette spezifiziert den Punkt, an dem ein Paket verarbeitet wird.

Die Option **-j *<target>*** legt das Ziel der Regel fest, also das Verhalten, wenn ein Paket mit einer Regel übereinstimmt. Beispiele für integrierte Ziele sind ACCEPT, DROP und REJECT.

Werfen Sie einen Blick auf die **iptables**-Handbuchseite für weitere Informationen über die verfügbaren Ketten, Optionen und Ziele.

2.5.3.2. Grundlegende Firewall-Richtlinien

Das Einrichten einfacher Firewall-Richtlinien kann als Grundlage für detailliertere, benutzerdefinierte Regeln dienen.

Jede **iptables**-Kette besteht aus einer Standardrichtlinie und null oder mehr Regeln, die zusammen mit der Standardrichtlinie das gesamte Regelset der Firewall definieren.

Die Standardrichtlinie für eine Kette ist entweder DROP oder ACCEPT. Sicherheitsbewusste Administratoren implementieren üblicherweise die Standardrichtlinie DROP und erlauben spezifische Pakete nur fallweise. Beispielsweise blockieren die folgenden Richtlinien alle eingehenden und ausgehenden Pakete auf einem Netzwerk-Gateway:

```
[root@myServer ~ ] # iptables -P INPUT DROP
[root@myServer ~ ] # iptables -P OUTPUT DROP
```

Es wird empfohlen, dass sämtliche *weitergeleiteten Pakete* — also Netzwerkverkehr, der von der Firewall an den Zielknoten geleitet wird — ebenfalls abgewiesen werden, um interne Clients davor zu bewahren, unbeabsichtigt dem Internet ausgesetzt zu werden. Verwenden Sie dazu die folgende Regel:

```
[root@myServer ~ ] # iptables -P FORWARD DROP
```

Nachdem Sie die Standardrichtlinien für alle Ketten festgelegt haben, können Sie neue Regeln für Ihre speziellen Netzwerk- und Sicherheitsbedürfnisse erstellen.

Die folgenden Abschnitte beschreiben das Speichern von iptables-Regeln sowie einige Beispiel für Regeln, die Sie beim Aufbau Ihrer iptables-Firewall implementieren können.

2.5.3.3. Speichern und Wiederherstellen von IPTables-Regeln

Änderungen an **iptables** sind nicht dauerhaft. Wenn das System oder der **iptables**-Dienst neu gestartet wird, werden die Regeln automatisch gelöscht und zurückgesetzt. Um die Regeln zu speichern und sie beim Start des **iptables**-Dienstes zu laden, führen Sie den folgenden Befehl aus:

```
[root@myServer ~ ] # service iptables save
```

Die Regeln werden in der **/etc/sysconfig/iptables**-Datei gespeichert und werden angewendet, sobald der Dienst oder der Rechner neu gestartet wird.

2.5.4. Häufige IPTables-Filter

Zu den wichtigsten Aspekten der Netzwerksicherheit gehört es, Angreifer von außerhalb am Zugriff auf ein LAN zu hindern. Die Integrität eines LAN sollte mithilfe einer Firewall vor böswilligen Benutzern von außerhalb geschützt werden.

Allerdings ist es mit einer Standardrichtlinie, die alle eingehenden, ausgehenden und weitergeleiteten Pakete blockiert, für die Firewall/das Gateway und interne LAN-Benutzer unmöglich, miteinander oder mit externen Ressourcen zu kommunizieren.

Um es Benutzern zu ermöglichen, Netzwerkfunktionen und Netzwerkapplikationen auszuführen, müssen Administratoren bestimmte Ports zur Kommunikation öffnen.

Um beispielsweise Zugriff auf Port 80 *auf der Firewall* zu erlauben, fügen Sie die folgende Regel hinzu:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Dies erlaubt es Benutzern, Websites zu besuchen, die über den Standardport 80 kommunizieren. Um Zugriff auf sichere Websites zu erlauben (z. B. https://www.example.com/), müssen Sie zudem den Zugriff auf Port 443 erlauben. Führen Sie dazu den folgenden Befehl aus:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Wichtig

Beim Erstellen eines **`iptables`**-Regelsets ist die Reihenfolge von entscheidender Bedeutung. Wenn eine Regel spezifiziert, dass alle Pakete vom 192.168.100.0/24 Subnetz verworfen werden, und darauf eine Regel folgt, die Pakete von 192.168.100.13 (was innerhalb des verworfenen Subnetz liegts) erlaubt, dann wird die zweite Regel ignoriert.
Die Regel, die Pakte von 192.168.100.13 erlaubt, muss sich in der Reihenfolge vor der Regel befinden, welche Pakete vom restlichen Subnetz verwirft.
Um eine Regel an einer bestimmten Stelle in eine vorhandene Kette einzufügen, verwenden Sie die **`-I`**-Option. Zum Beispiel:

```
[root@myServer ~ ] # iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

Diese Regel wird als erste Regel in die INPUT-Kette eingefügt, um lokalen Datenverkehr vom Loopback-Gerät zu erlauben.

In bestimmten Situationen benötigen Sie unter Umständen Zugriff auf das LAN von Remote aus. Um diese Remote-Verbindungen zu LAN-Diensten zu verschlüsseln, können Sie sichere Dienste wie z. B. SSH verwenden.

Administratoren mit PPP-basierten Ressourcen (wie z. B. Modembänke oder ISP-Accounts) können Einwählverbindungen nutzen, um sicher die Firewall-Barrieren zu umgehen. Da es sich bei Modemverbindungen um direkte Verbindungen handelt, sind diese üblicherweise hinter einer Firewall bzw. einem Gateway.

Für Remote-Benutzer mit Breitbandverbindungen können jedoch Sonderfälle eingerichtet werden. Sie können **`iptables`** konfigurieren, um Verbindungen von entfernten SSH-Clients zu akzeptieren. Die folgende Regel erlaubt beispielsweise SSH-Zugriff von Remote aus:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@myServer ~ ] # iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Diese Regeln erlauben eingehende und ausgehende Verbindungen für ein einzelnes System, wie z. B. ein einzelner PC, der direkt mit dem Internet oder einer Firewall bzw. einem Gateway verbunden ist. Allerdings erlauben diese Regeln Knoten hinter der Firewall bzw. dem Gateway nicht den Zugriff auf diese Dienste. Um LAN-Zugriff auf diese Dienste zu erlauben, können Sie *Network Address Translation* (NAT) mit **`iptables`** Filterungsregeln nutzen.

2.5.5. `FORWARD`- und NAT-Regeln

Die meisten Internet-Anbieter stellen ihren Unternehmenskunden nur eine begrenzte Anzahl an öffentlich routbaren IP-Adressen zur Verfügung.

Administratoren müssen daher andere Wege finden, um den Zugang auf Internetdienste gemeinsam zu verwenden, ohne an jeden Knoten auf dem LAN öffentliche IP-Adressen zu vergeben. Üblicherweise werden private IP-Adressen genutzt, um allen Knoten auf einem LAN den einwandfreien Zugriff auf interne und externe Netzwerkdienste zu ermöglichen.

Edge Router (wie z. B. Firewalls) können eingehende Daten vom Internet empfangen und die Pakete an den entsprechenden LAN-Knoten weiterleiten. Gleichzeitig können Firewalls/Gateways ausgehende Anfragen von einem LAN-Knoten an den entfernten Internet-Dienst weiterleiten.

Diese Weiterleitung von Netzwerkdaten kann jedoch unter Umständen gefährlich werden, insbesondere

angesichts moderner Cracking-Werkzeuge, die *interne* IP-Adressen ausschnüffeln können und es dem Angreifer so ermöglichen, sich als Knoten auf Ihrem LAN auszugeben.

Um dies zu verhindern, bietet **`iptables`** Richtlinien zur Um- und Weiterleitung von Paketen, die den untypischen Gebrauch von Netzwerkressourcen verhindern können.

Mithilfe der **`FORWARD`**-Kette kann ein Administrator steuern, wohin innerhalb eines LANs Pakete weitergeleitet werden können. Um beispielsweise die Weiterleitung für das gesamte LAN zu gestatten (vorausgesetzt, der Firewall bzw. dem Gateway ist eine interne IP-Adresse auf eth1 zugewiesen), verwenden Sie die folgenden Regeln:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth1 -j ACCEPT
[root@myServer ~ ] # iptables -A FORWARD -o eth1 -j ACCEPT
```

Diese Regeln ermöglichen Systemen hinter der Firewall bzw. dem Gateway Zugriff auf das interne Netzwerk. Das Gateway leitet Pakete von einem LAN-Knoten an den gewünschten Zielknoten weiter und leitet dabei alle Pakete durch sein **`eth1`**-Gerät.

Anmerkung

Standardmäßig deaktiviert die IPv4-Richtlinie in Red Hat Enterprise Linux Kernels die Unterstützung für IP-Forwarding. Dadurch ist es Rechnern, auf denen Red Hat Enterprise Linux läuft, nicht möglich, als dedizierte Edge-Router zu fungieren. Um IP-Forwarding zu aktivieren, führen Sie den folgenden Befehl aus:

```
[root@myServer ~ ] # sysctl -w net.ipv4.ip_forward=1
```

Diese Konfigurationsänderung gilt nur für die aktuelle Sitzung, sie ist nicht über einen Systemneustart oder Netzwerkneustart hinweg persistent. Um das IP-Forwarding dauerhaft zu aktivieren, bearbeiten Sie die **`/etc/sysctl.conf`**-Datei wie folgt:
Suchen Sie die folgende Zeile:

```
net.ipv4.ip_forward = 0
```

Bearbeiten Sie sie wie folgt:

```
net.ipv4.ip_forward = 1
```

Verwenden Sie den folgenden Befehl, um die Änderung an der **`sysctl.conf`**-Datei zu aktivieren:

```
[root@myServer ~ ] # sysctl -p /etc/sysctl.conf
```

2.5.5.1. Postrouting und IP Masquerading

Indem weitergeleitete Pakete über das interne IP-Gerät der Firewall akzeptiert werden, wird LAN-Knoten zwar die Kommunikation untereinander ermöglicht. Allerdings können sie nach wie vor nicht extern mit dem Internet kommunizieren.

Um LAN-Knoten mit privaten IP-Adressen die Kommunikation mit externen öffentlichen Netzwerken zu ermöglichen, konfigurieren Sie die Firewall für *IP-Masquerading*. Dies maskiert Anfragen der LAN-Knoten mit der IP-Adresse des externen Geräts der Firewall (in diesem Fall eth0):

```
[root@myServer ~ ] # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Diese Regel nutzt die NAT-Tabelle (**-t nat**) und spezifiziert die integrierte POSTROUTING-Kette für NAT (**-A POSTROUTING**) auf dem externen Netzwerkgerät der Firewall (**-o eth0**).

POSTROUTING ermöglicht die Veränderung von Paketen, wenn diese das externe Gerät der Firewall verlassen.

Das Ziel **-j MASQUERADE** wird spezifiziert, um die private IP-Adresse eines Knotens mit der externen IP-Adresse der Firewall bzw. des Gateways zu maskieren.

2.5.5.2. Prerouting

Falls Sie einen Server auf Ihrem internen Netzwerk haben, den Sie extern zugänglich machen möchten, können Sie das Ziel **-j DNAT** der PREROUTING-Kette in NAT verwenden, um eine Ziel-IP-Adresse und einen Port anzugeben, an die eingehende Pakete, die eine Verbindung mit Ihrem internen Dienst anfragen, weitergeleitet werden können.

Um beispielsweise eingehende HTTP-Anfragen an Ihren dedizierten Apache HTTP Server unter 172.31.0.23 weiterzuleiten, verwenden Sie den folgenden Befehl:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
DNAT --to 172.31.0.23:80
```

Diese Regel legt fest, dass die nat-Tabelle die integrierte PREROUTING-Kette verwendet, um eingehende HTTP-Anfragen exklusiv an die aufgeführte Ziel-IP-Adresse 172.31.0.23 weiterzuleiten.

Anmerkung

Falls Sie in Ihrer FORWARD-Kette die Standardrichtlinie DROP verwenden, müssen Sie eine Regel anhängen, die alle eingehenden HTTP-Anfragen weiterleitet, so dass das Ziel-NAT-Routing ermöglicht wird. Führen Sie dazu den folgenden Befehl aus:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth0 -p tcp --dport 80 -d
172.31.0.23 -j ACCEPT
```

Die Regel leitet alle eingehenden HTTP-Anfragen von der Firewall an das vorgesehene Ziel weiter, den Apache HTTP-Server hinter der Firewall.

2.5.5.3. DMZs und IPTables

Sie können **iptables**-Regeln erstellen, um Daten an bestimmte Rechner weiterzuleiten, wie z. B. an dedizierte HTTP- oder FTP-Server, in einer *demilitarisierten Zone* (DMZ). Eine DMZ ist ein spezielles, lokales Subnetzwerk, das Dienste auf einem öffentlichen Träger wie z. B. dem Internet bereitstellt.

Um beispielsweise eine Regel zur Weiterleitung von eingehenden HTTP-Anfragen an einen dedizierten HTTP-Server unter 10.0.4.2 (außerhalb des 192.168.1.0/24 Bereichs des LANs) weiterzuleiten, verwendet NAT die **PREROUTING**-Tabelle, um Pakete an das entsprechende Ziel weiterzuleiten:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
DNAT --to-destination 10.0.4.2:80
```

Mit diesem Befehl werden alle HTTP-Verbindungen zu Port 80 von außerhalb des LANs an den HTTP-

Server auf einem vom Rest des internen Netzwerks getrennten Netzwerk geleitet. Diese Art der Netzwerksegmentierung kann sicherer sein, als HTTP-Verbindungen zu einem Rechner auf dem Netzwerk zu erlauben.

Falls der HTTP-Server konfiguriert ist, um sichere Verbindungen zu akzeptieren, muss auch Port 443 weitergeleitet werden.

2.5.6. Schädliche Software und erschnüffelte IP-Adressen

Sie können auch ausgeklügeltere Regeln erstellen, die den Zugriff auf bestimmte Subnetze oder gar bestimmte Knoten innerhalb eines LANs regeln. Auch können Sie bestimmte, zweifelhafte Applikationen oder Programme wie z. B. Trojaner, Würmer und andere Client-/Server-Viren daran hindern, Verbindungen zu deren Servern herzustellen.

Beispielsweise scannen einige Trojaner Netzwerke nach Diensten auf Ports 31337 bis 31340 (in Cracking-Terminologie auch *Elite*-Ports genannt).

Da es keine legitimen Dienste gibt, die auf diesen nicht-standardmäßigen Ports kommunizieren, können Sie durch deren Sperrung das Risiko mindern, dass potenziell infizierte Knoten auf Ihrem Netzwerk selbstständig mit ihren entfernten Master-Servern kommunizieren.

Die folgenden Regeln verwerfen jeglichen TCP-Datenverkehr, der Port 31337 zu benutzen versucht:

```
[root@myServer ~ ] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport
31337 -j DROP
[root@myServer ~ ] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport
31337 -j DROP
```

Sie können auch Verbindungen von außerhalb blockieren, die versuchen, private IP-Adressbereiche zu erschnüffeln, um damit Ihr LAN infiltrieren zu können.

Wenn Ihr LAN zum Beispiel den Bereich 192.168.1.0/24 verwendet, können Sie eine Regel aufstellen, die das mit dem Internet verbundene Gerät (z. B. eth0) anweist, alle Pakete an dieses Gerät zu verwerfen, die eine IP-Adresse innerhalb des Bereichs Ihres LANs haben.

Da als Standardrichtlinie empfohlen wird, weitergeleitete Pakete zurückzuweisen, werden sämtliche andere erschnüffelte IP-Adressen zum externen Gerät (eth0) automatisch zurückgewiesen.

```
[root@myServer ~ ] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

Anmerkung

Beim *Anfügen* von Regeln gibt es Unterschiede zwischen den **`DROP`**- und **`REJECT`**-Zielen. Das **`REJECT`**-Ziel verweigert den Zugang und gibt eine **`connection refused`** Fehlermeldung an Benutzer heraus, die mit dem Dienst zu verbinden versuchen. Das **`DROP`**-Ziel verwirft das Paket dagegen ohne jegliche Fehlermeldung.
Administratoren können diese Ziele nach eigenem Ermessen einsetzen. Allerdings ist es empfehlenswert, dass **`REJECT`**-Ziel zu verwenden, um Benutzer nicht unnötig zu verwirren und um wiederholte Verbindungsversuche zu vermeiden.

2.5.7. IPTables und Connection Tracking

Sie können Verbindungen zu Diensten untersuchen und basierend auf deren *Verbindungszustand* einschränken. Ein Modul innerhalb von **`iptables`** verwendet eine Methode, die *Connection Tracking*

oder auch *Dynamische Paketfilterung* genannt wird, um Informationen über eingehende Verbindungen zu speichern. Sie können den Zugriff auf Grundlage der folgenden Verbindungszustände erlauben oder verweigern:

- **NEW** — Ein Paket fordert eine neue Verbindung an, z. B. eine HTTP-Anfrage.
- **ESTABLISHED** — Ein Paket ist Teil einer bestehenden Verbindung.
- **RELATED** — Ein Paket fordert eine neue Verbindung an, ist jedoch Teil einer bestehenden Verbindung. So verwendet FTP beispielsweise Port 21, um eine Verbindung herzustellen, die Daten werden jedoch auf einem anderen Port übertragen (üblicherweise Port 20).
- **INVALID** — Ein Paket gehört zu keiner Verbindung in der Connection-Tracking-Tabelle.

Sie können die **iptables**-Funktion zur Zustandsüberprüfung mit jedem Netzwerkprotokoll verwenden, selbst wenn das Protokoll selbst zustandslos ist (wie z. B. UDP). Das folgende Beispiel zeigt eine Regel, die mithilfe der dynamischen Paketfilterung nur solche Pakete weiterleitet, die mit einer bereits bestehenden Verbindung zusammenhängen:

```
[root@myServer ~ ] # iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

2.5.8. IPv6

Die Einführung des Internet-Protokolls der nächsten Generation, IPv6 genannt, erweitert die Möglichkeiten des 32-Bit-Adressenlimits von IPv4 (oder IP). IPv6 unterstützt 128-Bit-Adressen, weshalb IPv6-kompatible Trägernetzwerke eine größere Anzahl routbarer Adressen ansprechen können als mit IPv4.

Red Hat Enterprise Linux unterstützt die IPv6 Firewall-Regeln unter Verwendung des Netfilter 6 Subsystems und des **ip6tables**-Befehls. In Red Hat Enterprise Linux 6 sind sowohl IPv4- als auch IPv6-Dienste standardmäßig aktiviert.

Die **ip6tables**-Befehlssyntax ist identisch mit **iptables**, mit Ausnahme der Tatsache, dass es 128-Bit-Adressen unterstützt. Führen Sie beispielsweise den folgenden Befehl aus, um SSH-Verbindungen auf einem IPv6-kompatiblen Netzwerkserver zu aktivieren:

```
[root@myServer ~ ] # ip6tables -A INPUT -i eth0 -p tcp -s 3ffe:ffff:100::1/128 --
dport 22 -j ACCEPT
```

Für mehr Information über IPv6-Networking werfen Sie bitte einen Blick auf die IPv6-Informationsseite unter http://www.ipv6.org/.

2.5.9. Zusätzliche Informationsquellen

Einige Aspekte von Firewalls und des Linux-Netfilter-Subsystems konnten in diesem Kapitel nicht abgedeckt werden. Für weitere Informationen ziehen Sie bitte die folgenden Quellen zu Rate:

2.5.9.1. Installierte Firewall-Dokumentation

- Abschnitt 2.6, „IPTables“ beinhaltet ausführliche Informationen über den **iptables**-Befehl, einschließlich der Definitionen für zahlreiche Befehlsoptionen.
- Die **iptables**-Handbuchseite enthält eine kurze Zusammenfassung der verschiedenen Optionen.

2.5.9.2. Hilfreiche Firewall-Websites

- http://www.netfilter.org/ — Die offizielle Homepage des Netfilter- und **iptables**-Projekts.

- http://www.tldp.org/ — Das Linux-Dokumentations-Projekt enthält mehrere hilfreiche Handbücher in Zusammenhang mit der Erstellung und Administration von Firewalls.
- http://www.iana.org/assignments/port-numbers — Die offizielle Liste registrierter und üblicher Dienst-Ports, zugeteilt von der Internet Assigned Numbers Authority.

2.5.9.3. Verwandte Dokumentation

- *Red Hat Linux Firewalls*, von Bill McCarty; Red Hat Press — ein umfassendes Nachschlagewerk zum Erstellen von Netzwerk- und Server-Firewalls mittels Open-Source-Paketfilterungs-Technologie wie z. B. Netfilter und **iptables**. Es beinhaltet Themen wie beispielsweise das Analysieren von Firewall-Protokollen, das Entwickeln von Firewall-Regeln und das Anpassen Ihrer Firewall mit grafischen Tools wie z. B. **lokkit**.
- *Linux Firewalls*, von Robert Ziegler; New Riders Press — enthält eine Menge an Informationen über das Erstellen von Firewalls mit 2.2 Kernel und **ipchains** sowie mit Netfilter und **iptables**. Es werden auch zusätzliche Sicherheitsthemen behandelt, wie z. B. der Fernzugriff und Intrusion-Detection-Systeme.

2.6. IPTables

Red Hat Enterprise Linux beinhaltet erweiterte Tools für die *Paketfilterung* — den Prozess zur Kontrolle von Netzwerkpaketen, während diese den Netzwerkstapel des Kernels durchlaufen. Die Kernel-Versionen vor 2.4 nutzten **ipchains** zur Paketfilterung und wendeten Regellisten auf Pakete in jeder Phase des Filterungsprozesses an. Mit der Kernel-Version 2.4 wurde **iptables** eingeführt (auch *Netfilter* genannt), die den **ipchains** zwar ähnlich sind, jedoch den Wirkungsbereich und die Kontrollmöglichkeiten bei der Filterung von Netzwerkpaketen stark erweitern.

Dieses Kapitel behandelt die Grundlagen der Paketfilterung und erläutert die verschiedenen, verfügbaren Optionen für die **iptables**-Befehle. Es wird außerdem gezeigt, wie Filterungsregeln über Neustarts des Systems hinweg bewahrt werden können.

Unter Abschnitt 2.6.6, „Zusätzliche Informationsquellen“ finden Sie Anweisungen, wie Sie **iptables**-Regeln angelegen und darauf basierend eine Firewall einrichten können.

Wichtig

In Kernel-Versionen 2.4 und höher ist der standardmäßige Firewall-Mechanismus zwar **iptables**, allerdings kann **iptables** nicht benutzt werden, wenn **ipchains** bereits läuft. Falls zum Zeitpunkt des Systemstarts **ipchains** bereits vorhanden ist, gibt der Kernel eine Fehlermeldung aus und kann **iptables** nicht starten.
Die Funktionalität von **ipchains** wird durch diese Fehlermeldungen jedoch nicht beeinträchtigt.

2.6.1. Paketfilterung

Der Linux-Kernel verwendet die **Netfilter**-Facility, um Pakete zu filtern, wodurch dem System das Empfangen oder Weiterleiten einiger der Pakete erlaubt wird, während andere Pakete gestoppt werden. Diese Facility ist im Linux-Kernel integriert und enthält die folgenden drei *Tabellen* oder *Regellisten*:

- **filter** — Die Standardtabelle zur Verarbeitung von Netzwerkpaketen.
- **nat** — Diese Tabelle wird zur Änderung von Paketen verwendet, die eine neue Verbindung herstellen, sowie für *Network Address Translation* (*NAT*).
- **mangle** — Diese Tabelle wird für spezielle Arten der Paketänderung verwendet.

Jede dieser Tabellen verfügt über eine Reihe integrierter *Ketten* (engl: chains), die den Aktionen entsprechen, die von **`netfilter`** auf dem Paket durchgeführt werden.

Die integrierten Ketten für die **`filter`**-Tabelle sind:

- *INPUT* — Gilt für Netzwerkpakete, die für den Host bestimmt sind.
- *OUTPUT* — Gilt für Netzwerkpakete, die lokal generiert wurden.
- *FORWARD* — Gilt für Netzwerkpakete, die über den Host geroutet werden.

Die integrierten Ketten für die **`nat`**-Tabelle sind:

- *PREROUTING* — Ändert Netzwerkpakete beim Empfang.
- *OUTPUT* — Ändert lokal generierte Netzwerkpakete, bevor diese gesendet werden.
- *POSTROUTING* — Ändert Netzwerkpakete, bevor diese gesendet werden.

Die integrierten Ketten für die **`mangle`**-Tabelle sind:

- *INPUT* — Ändert Netzwerkpakete, die für den Host bestimmt sind.
- *OUTPUT* — Ändert lokal generierte Netzwerkpakete, bevor diese gesendet werden.
- *FORWARD* — Ändert Netzwerkpakete, die über den Host geroutet werden.
- *PREROUTING* — Ändert eingehende Netzwerkpakete, bevor diese geroutet werden.
- *POSTROUTING* — Ändert Netzwerkpakete, bevor diese gesendet werden.

Jedes Netzwerkpaket, das von einem Linux-System empfangen oder ausgesendet wird, fällt mindestens unter eine dieser Tabellen. Ein Paket kann jedoch in jeder Tabelle auf mehrere Regeln hin überprüft werden, bevor es am Ende der Kette wieder austritt. Struktur und Zweck dieser Regeln können unterschiedlich sein, in der Regel versuchen sie jedoch ein Paket zu identifizieren, das von einer bzw. an eine bestimmte IP-Adresse gesendet wurde, wenn dieses ein bestimmtes Protokoll und einen bestimmten Netzwerkdienst benutzt. Die folgende Abbildung veranschaulicht, wie der Durchlauf der Pakete vom IPTables-Subsystem untersucht wird:

Anmerkung

Standardmäßig werden Firewall-Regeln in den Dateien **`/etc/sysconfig/iptables`** oder **`/etc/sysconfig/ip6tables`** gespeichert.
Der **`iptables`**-Dienst startet beim Booten eines Linux-Systems vor jeglichen DNS-Diensten. Aus diesem Grund können Firewall-Regeln nur auf numerische IP-Adressen (zum Beispiel 192.168.0.1) verweisen. Domainnamen (wie beispielsweise host.example.com) in solchen Regeln verursachen dagegen Fehler.

Sobald Pakete mit einer bestimmten Regel in einer der Tabellen übereinstimmen, wird unabhängig von ihrem Bestimmungsort ein *Ziel* bzw. eine Aktion auf sie angewendet. Falls die Regel ein **`ACCEPT`**-Ziel für ein übereinstimmendes Paket spezifiziert, überspringt das Paket die restlichen Regeln und darf somit seinen Weg zum Bestimmungsort fortsetzen. Wenn aber eine Regel ein **`DROP`**-Ziel spezifiziert, wird dem Paket der Zugriff auf das System verwehrt, ohne eine Meldung an den Host-Rechner, von dem das Paket stammt, zurückzusenden. Wenn eine Regel ein **`QUEUE`**-Ziel spezifiziert, wird das Paket an den Userspace weitergeleitet. Wenn eine Regel ein optionales **`REJECT`**-Ziel spezifiziert, wird das Paket verworfen und es wird ein Fehlerpaket an den Ursprungs-Host zurückgesendet.

Jede Kette hat eine Standardrichtlinie, entweder **`ACCEPT`**, **`DROP`**, **`REJECT`** oder **`QUEUE`**. Wenn das Paket keiner der Regeln in der Kette entspricht, wird auf dieses Paket die Standardrichtlinie angewendet.

Der **`iptables`**-Befehl konfiguriert diese Tabellen und erstellt neue, falls nötig.

2.6.2. Befehlsoptionen für IPTables

Regeln zum Filtern von Paketen werden mithilfe des **`iptables`**-Befehls erstellt. Die folgenden Aspekte eines Pakets werden häufig als Kriterien verwendet:

- *Pakettyp* — Legt fest, welche Pakete der Befehl filtert, basierend auf dem Typ des Pakets.
- *Paketquelle/-bestimmungort* — Legt fest, welche Pakete der Befehl filtert, basierend auf der Quelle oder dem Bestimmungort des Pakets.
- *Ziel* — Legt fest, welche Aktion auf den Paketen angewendet wird, die den oben genannten Kriterien entsprechen.

Werfen Sie einen Blick auf Abschnitt 2.6.2.4, „IPTables Übereinstimmungsoptionen“ und Abschnitt 2.6.2.5, „Zieloptionen“ für weitere Informationen über bestimmte Optionen, die diese Aspekte eines Pakets betreffen.

Die mit bestimmten **`iptables`**-Regeln verwendeten Optionen müssen logisch gruppiert sein, basierend auf Zweck und Bedingungen der gesamten Regel, damit die Regel gültig ist. Der Rest dieses Abschnitts erläutert häufig verwendete Optionen für den **`iptables`**-Befehl.

2.6.2.1. Syntax der IPTables-Befehlsoptionen

Viele **`iptables`**-Befehle folgen dem folgenden Format:

```
 iptables [-t <table-name>] <command> <chain-name> \ <parameter-1> <option-1> \
<parameter-n> <option-n>
```

<table-name> — Legt fest, auf welche Tabelle sich diese Regel bezieht. Falls nichts angegeben ist, wird die **`filter`**-Tabelle verwendet.

<command> — Legt fest, welche Aktion durchgeführt werden soll, z. B. Hinzufügen oder Löschen einer

Regel.

<chain-name> — Legt die Kette fest, die bearbeitet, erstellt oder gelöscht werden soll.

<parameter>-<option> Paare — Parameter und zugehörige Optionen, die festlegen, wie ein Paket zu verarbeiten ist, auf das diese Regel zutrifft.

Die Länge und Komplexität eines **iptables**-Befehls kann sich sehr unterscheiden, abhängig von dessen Zweck.

Beispielsweise kann ein Befehl zum Löschen einer Regel aus einer Kette sehr kurz sein:

```
iptables -D <chain-name> <line-number>
```

Dagegen kann ein Befehl, der eine Regel hinzufügt, die Pakete von einem bestimmten Subnetz anhand einer Vielzahl an speziellen Parametern und Optionen filtert, ziemlich lang sein. Beim Zusammensetzen des **iptables**-Befehls muss bedacht werden, dass einige Parameter und Optionen weitere Parameter und Optionen benötigen, um eine gültige Regel zu bilden. Dies kann einen Dominoeffekt hervorrufen, mit weiteren Parametern, die wiederum weitere Parameter benötigen. Solange nicht alle Parameter und Optionen, die eine Reihe weiterer Optionen benötigen, erfüllt sind, ist die Regel nicht gültig.

Wenn Sie **iptables -h** eingeben, erhalten Sie eine vollständige Liste der **iptables**-Befehlsstrukturen.

2.6.2.2. Befehlsoptionen

Mithilfe von Befehlsoptionen wird **iptables** angewiesen, einen bestimmten Vorgang auszuführen. Nur eine einzige Befehlsoption pro **iptables**-Befehl ist erlaubt. Mit Ausnahme des Hilfebefehls sind alle Befehle in Großbuchstaben geschrieben.

Die **iptables**-Befehle sind:

- **-A** — Hängt die Regel an das Ende der angegebenen Kette an. Im Gegensatz zur weiter unten beschriebenen Option **-I** wird hierbei kein ganzzahliger Parameter verwendet. Die Regel wird immer an das Ende der angegebenen Kette gehängt.
- **-D <integer> | <rule>** — Entfernt eine Regel in einer bestimmten Kette anhand ihrer Nummer (z. B. **5** für die fünfte Regel einer Kette) oder durch Angabe einer Regelspezifikation. Die Regelspezifikation muss exakt mit einer bestehenden Regel übereinstimmen.
- **-E** — Benennt eine benutzerdefinierte Kette um. Eine benutzerdefinierte Kette ist jede Kette, die nicht eine standardmäßige, voreingestellte Kette ist. (Werfen Sie einen Blick auf die Option **-N** weiter unten für Informationen zur Erstellung von benutzerdefinierten Ketten). Dies ist eine reine Schönheitskorrektur und beeinflusst nicht die Struktur der Tabelle.

Anmerkung

Falls Sie versuchen, eine der Standardketten umzubenennen, gibt das System die Fehlermeldung **Match not found** aus. Sie können die Standardketten nicht umbenennen.

- **-F** — Löscht den Inhalt der gewählten Kette, woraufhin effektiv jede Regel in der Kette entfernt wird. Wenn keine Kette angegeben wird, löscht dieser Befehl jede Regel in jeder Kette.
- **-h** — Liefert eine Liste mit Befehlsstrukturen sowie eine kurze Zusammenfassung der Befehlsparameter und -optionen.
- **-I [<integer>]** — Fügt eine Regel an einem bestimmten Punkt, der anhand eines ganzzahligen,

benutzerdefinierten Werts spezifiziert wird, in eine Kette ein. Wird kein Wert angegeben, wird die Regel am Anfang der Kette eingefügt.

Wichtig

Wie bereits oben erwähnt, bestimmt die Reihenfolge der Regeln in einer Kette, welche Regeln auf welche Pakete angewendet werden. Dies sollten Sie beim Hinzufügen von Regeln mit der Option **-A** oder **-I** unbedingt bedenken.
Dies ist besonders wichtig, wenn Regeln unter Verwendung der Option **-I** mit einem ganzzahligen Parameter hinzugefügt werden. Wenn Sie beim Hinzufügen einer Regel zu einer Kette eine bereits existierende Nummer angeben, fügt **iptables** die neue Regel *vor* (also über) der existierenden Regel ein.

- **-L** — Listet alle Regeln in der angegebenen Kette auf. Um alle Regeln in allen Ketten in der Standardtabelle **filter** aufzulisten, spezifizieren Sie keine Kette oder Tabelle. Ansonsten sollte folgende Syntax verwendet werden, um die Regeln in einer bestimmten Kette in einer bestimmten Tabelle aufzulisten:

```
iptables -L <chain-name> -t <table-name>
```

  Zusätzliche Optionen für die **-L**-Befehlsoption, die z. B. Regelnummern anzeigen oder ausführlichere Regelbeschreibungen ermöglichen, finden Sie in Abschnitt 2.6.2.6, „Auflistungsoptionen“.
- **-N** — Erstellt eine neue Kette mit benutzerdefiniertem Namen. Der Name der Kette muss eindeutig sein, andernfalls wird eine Fehlermeldung angezeigt.
- **-P** — Legt die Standardrichtlinie für die angegebene Kette fest. Dadurch werden Pakete, die eine Kette vollständig durchlaufen, ohne mit einer Regel übereinzustimmen, an das angegebene Ziel gesendet, wie z. B. ACCEPT oder DROP.
- **-R** — Ersetzt eine Regel in einer angegebenen Kette. Sie müssen dazu nach dem Namen der Kette eine Regelnummer angeben, um die Regel zu ersetzen. Die erste Regel einer Kette ist die Regel Nummer 1.
- **-X** — Löscht eine benutzerdefinierte Kette. Eine integrierte Kette kann dagegen nicht gelöscht werden.
- **-Z** — Stellt Byte- und Paketzähler in allen Ketten für eine Tabelle auf Null.

2.6.2.3. IPTables-Parameteroptionen

Bestimmte **iptables**-Befehle, zum Beispiel die Befehle zum Hinzufügen, Anhängen, Entfernen, Einfügen oder Ersetzen von Regeln innerhalb einer bestimmten Kette, erfordern verschiedene Parameter für die Erstellung einer Paketfilterungsregel.

- **-c** — Setzt die Zähler für eine angegebene Regel zurück. Dieser Parameter akzeptiert die **PKTS**- und **BYTES**-Optionen um anzugeben, welche Zähler zurückzusetzen sind.
- **-d** — Legt den Bestimmungsort des Pakets als Hostnamen, IP-Adresse oder Netzwerk fest, der mit der Regel übereinstimmt. Zur Übereinstimmung mit einem Netzwerk werden die folgenden Formate für IP-Adressen/Netmasks unterstützt:
  - ***N.N.N.N/M.M.M.M*** — Wobei ***N.N.N.N*** der IP-Adressbereich und ***M.M.M.M*** die Netmask ist.
  - ***N.N.N.N/M*** — Wobei ***N.N.N.N*** der IP-Adressbereich und ***M*** die Bitmask ist.
- **-f** — Wendet diese Regel nur auf fragmentierte Pakete an.

  Wird ein Ausrufezeichen (**!**) als Option vor diesem Parameter verwendet, werden nur unfragmentierte Pakete verglichen.

Anmerkung

Die Unterscheidung zwischen fragmentierten und unfragmentierten Paketen ist wünschenswert, auch wenn fragmentierte Pakete standardmäßig Teil des IP-Protokolls sind. Fragmentierung wurde ursprünglich dazu konzipiert, IP-Paketen das Passieren von Netzwerken mit unterschiedlichen Frame-Größen zu gestatten, wird heutzutage jedoch meist dazu verwendet, mithilfe von fehlerhaften Paketen DoS-Angriffe zu unternehmen. An dieser Stelle sei auch erwähnt, dass IPv6 Fragmentierung komplett verbietet.

- **`-i`** — Legt die Eingangsnetzwerkschnittstelle fest, z. B. **`eth0`** oder **`ppp0`**. Mit **`iptables`** darf dieser zusätzliche Parameter nur mit INPUT- und FORWARD-Ketten in Verbindung mit der **`filter`**-Tabelle und der PREROUTING-Kette mit den **`nat`**- und **`mangle`**-Tabellen verwendet werden.

  Dieser Parameter unterstützt darüberhinaus folgende spezielle Optionen:

  - Ausrufezeichen (**`!`**) — Kehrt die Direktive um, was bedeutet, dass jegliche angegebene Schnittstellen von dieser Regel ausgenommen sind.
  - Pluszeichen (**`+`**) — Ein Platzhalterzeichen, mithilfe dessen alle Schnittstellen zutreffend sind, die mit der angegebenen Zeichenkette übereinstimmen. Der Parameter **`-i eth+`** würde diese Regel z. B. für alle Ethnernet-Schnittstellen Ihres Systems anwenden, aber alle anderen Schnittstellen, wie z. B. **`ppp0`** auslassen.

  Wenn der **`-i`**-Parameter ohne Angabe einer Schnittstelle verwendet wird, ist jede Schnittstelle von dieser Regel betroffen.

- **`-j`** — Springt zum angegebenen Ziel, wenn ein Paket einer bestimmten Regel entspricht.

  Die Standardziele sind **`ACCEPT`**, **`DROP`**, **`QUEUE`** und **`RETURN`**.

  Erweiterte Optionen sind zudem über Module verfügbar, die standardmäßig mit mit dem Red Hat Enterprise Linux **`iptables`**-RPM-Paket geladen werden. Gültige Ziele in diesen Modulen sind unter anderem **`LOG`**, **`MARK`** und **`REJECT`**. Weitere Informationen zu diesen und anderen Zielen finden Sie auf der **`iptables`**-Handbuchseite.

  Diese Option kann dazu verwendet werden, ein Paket, das einer bestimmten Regel entspricht, an eine benutzerdefinierte Kette außerhalb der aktuellen Kette weiterzuleiten, so dass andere Regeln auf dieses Paket angewendet werden können.

  Falls kein Ziel festgelegt ist, durchläuft das Paket diese Regel, ohne dass etwas unternommen wird. Der Zähler für diese Regel wird jedoch um eins erhöht.

- **`-o`** — Legt die Ausgangsnetzwerkschnittstelle für eine bestimmte Regel fest, die nur mit OUTPUT- und FORWARD-Ketten in der **`filter`**-Tabelle und mit der POSTROUTING-Kette in den **`nat`**- und **`mangle`**-Tabellen verwendet werden kann. Die akzeptierten Optionen dieses Parameters sind dieselben wie die des Parameters der Eingangsnetzwerkschnittstelle (**`-i`**).

- **`-p <protocol>`** — Legt das IP-Protokoll für die Regel fest. Dies kann entweder **`icmp`**, **`tcp`**, **`udp`**, **`all`**, oder aber ein numerischer Wert sein, der für eines dieser Protokolle oder für ein anderes Protokoll steht. Auch Protokolle, die in **`/etc/protocols`** aufgelistet sind, können verwendet werden.

  Die Option "**`all`**" bewirkt, dass die Regel auf alle unterstützte Protokolle angewendet wird. Falls kein Protokoll in der Regel angegeben wird, ist der Standardwert "**`all`**".

- **`-s`** — Legt die Quelle eines bestimmten Pakets fest, und zwar unter Verwendung derselben Syntax, die auch der Parameter für den Bestimmungsort (**`-d`**) verwendet.

2.6.2.4. IPTables Übereinstimmungsoptionen

Verschiedene Netzwerkprotokolle bieten verschiedene spezielle Übereinstimmungsoptionen, die konfiguriert werden können, um auf bestimmte Pakete zuzutreffen, die diese Protokolle verwenden. Das

Protokoll muss jedoch zuerst im **`iptables`**-Befehl spezifiziert werden. So aktiviert **`-p`** ***<protocol-name>*** z. B. Optionen für das angegebene Protokoll. Beachten Sie, dass Sie auch die Protokoll-ID anstelle des Protokollnamens verwenden können. Werfen Sie einen Blick auf die folgenden Beispiele, die jeweils denselben Effekt haben:

```
iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
```

```
iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

Die Definition von Diensten wird in der Datei **`/etc/services`** geliefert. Im Interesse der Lesbarkeit wird die Verwendung der Dienstnamen anstelle der Portnummern empfohlen.

Warnung

Sichern Sie die **`/etc/services`**-Datei, um ein unerlaubtes Bearbeiten zu verhindern. Ist diese Datei editierbar, können Angreifer sie dazu missbrauchen, Ports auf Ihrem Rechner zu aktivieren, die Sie geschlossen hatten. Um diese Datei abzusichern, führen Sie als Root folgende Befehle aus:

```
[root@myServer ~]# chown root.root /etc/services
[root@myServer ~]# chmod 0644 /etc/services
[root@myServer ~]# chattr +i /etc/services
```

Auf diese Weise wird verhindert, dass die Datei umbenannt oder gelöscht wird, bzw. Links zu ihr erstellt werden.

2.6.2.4.1. TCP-Protokoll

Folgende Übereinstimmungsoptionen stehen für das TCP-Protokoll zur Verfügung (**`-p tcp`**):

- **`--dport`** — Definiert den Ziel-Port für das Paket.

  Verwenden Sie den Namen eines Netzwerkdienstes (wie z. B. www oder smtp), eine Portnummer oder einen Bereich von Portnummern, um diese Option zu konfigurieren.

  Um einen Bereich von Portnummern anzugeben, trennen Sie die zwei Nummern durch einen Doppelpunkt (**`:`**), z. B.: **`-p tcp --dport 3000:3200`**. Der größtmögliche Bereich ist **`0:65535`**.

  Sie können auch ein Ausrufezeichen (**`!`**) vor der **`--dport`**-Option verwenden, um mit allen Paketen, die *nicht* diesen Netzwerkdienst oder diesen Port verwenden, übereinzustimmen.

  Um die Namen und Aliasse von Netzwerkdiensten und den von ihnen verwendeten Portnummern zu durchsuchen, werfen Sie einen Blick auf die **`/etc/services`**-Datei.

  Die Übereinstimmungsoption **`--destination-port`** ist dasselbe wie **`--dport`**.

- **`--sport`** — Setzt den Ursprungsport des Pakets unter Verwendung derselben Optionen wie **`--dport`**. Die Übereinstimmungsoption **`--source-port`** ist dasselbe wie **`--sport`**.

- **`--syn`** — Gilt für alle TCP-Pakete, die eine Kommunikation initialisieren sollen, allgemein *SYN-Pakete* genannt. Alle Pakete, die Nutzdaten enthalten, werden nicht bearbeitet.

  Wird ein Ausrufezeichen (**`!`**) vor der **`--syn`**-Option verwendet, wird die Regel nur auf Pakete angewendet, bei denen es sich nicht um SYN-Pakete handelt.

- **`--tcp-flags <tested flag list> <set flag list>`** — Ermöglicht TCP-Paketen mit bestimmten Bits (Flags), mit einer Regel übereinzustimmen.

  Die Übereinstimmungsoption **`--tcp-flags`** akzeptiert zwei Parameter. Beim ersten Parameter

handelt es sich um eine Maske, eine kommagetrennte Liste mit Flags, die im Paket zu untersuchen sind. Der zweite Parameter ist eine kommagetrennte Liste mit Flags, die gesetzt sein müssen, um eine Übereinstimmung mit der Regel zu erhalten.

Mögliche Flags sind:

- **ACK**
- **FIN**
- **PSH**
- **RST**
- **SYN**
- **URG**
- **ALL**
- **NONE**

Eine **iptables**-Regel, die folgende Spezifikation enthält, trifft beispielsweise nur auf TCP-Pakete zu, in denen das SYN-Flag aktiviert und die ACK- und FIN-Flags deaktiviert sind:

--tcp-flags ACK,FIN,SYN SYN

Verwenden Sie das Ausrufezeichen (**!**) hinter **--tcp-flags**, um den Effekt der Übereinstimmungsoption umzukehren.

- **--tcp-option** — Versucht eine Übereinstimmung anhand von TCP-spezifischen Optionen, die innerhalb eines bestimmten Pakets eingestellt werden können. Diese Übereinstimmungsoption kann ebenfalls mit dem Ausrufezeichen (**!**) umgekehrt werden.

2.6.2.4.2. UDP-Protokoll

Für das UDP-Protokoll stehen folgende Übereinstimmungsoptionen zur Verfügung (**-p udp**):

- **--dport** — Spezifiziert den Ziel-Port des UDP-Pakets unter Verwendung von Dienstnamen, Portnummer oder einem Portnummernbereich. Die Übereinstimmungsoption **--destination-port** ist dasselbe wie **--dport**.
- **--sport** — Spezifiziert den Ursprungs-Port des UDP-Pakets unter Verwendung von Dienstnamen, Portnummer oder einem Portnummernbereich. Die Übereinstimmungsoption **--source-port** ist dasselbe wie **--sport**.

Um einen spezifischen Portnummernbereich für die Optionen **--dport** und **--sport** anzugeben, trennen Sie die zwei Nummern durch einen Doppelpunkt (:), z. B.: **-p tcp --dport 3000:3200**. Der größtmögliche Bereich ist 0:65535.

2.6.2.4.3. ICMP-Protokoll

Die folgenden Übereinstimmungsoptionen sind für das Internet Control Message Protocol (ICMP) (**-p icmp**) verfügbar:

- **--icmp-type** — Bestimmt den Namen oder die Nummer des ICMP-Typs, der mit der Regel übereinstimmen soll. Durch Eingabe des Befehls **iptables -p icmp -h** wird eine Liste aller gültigen ICMP-Namen angezeigt.

2.6.2.4.4. Module mit zusätzlichen Übereinstimmungsoptionen

Zusätzliche Übereinstimmungsoptionen stehen durch Module zur Verfügung, die vom Befehl **iptables** geladen werden können.

Um ein Modul für Übereinstimmungsoptionen zu verwenden, müssen Sie das Modul namentlich mithilfe der Option **-m *<module-name>*** laden (wobei ***<module-name>*** der Name des Moduls ist).

Viele Module stehen standardmäßig zur Verfügung. Sie können zudem Ihre eigenen Module erstellen, um die Funktionalität zu erweitern.

Sehen Sie nachfolgend einige der am häufigsten verwendeten Module:

- **`limit`**-Modul — Schränkt die Anzahl der Pakete ein, auf die eine bestimmte Regel zutrifft.

  Wenn das **`limit`**-Modul in Verbindung mit dem **`LOG`**-Ziel verwendet wird, kann es verhindern, dass eine Flut übereinstimmender Pakete das Systemprotokoll mit sich wiederholenden Nachrichten überschwemmen bzw. Systemressourcen verbrauchen.

  Werfen Sie einen Blick auf Abschnitt 2.6.2.5, „Zieloptionen“ für weitere Informationen zum **`LOG`**-Ziel.

  Das **`limit`**-Modul akzeptiert die folgenden Optionen:

  - **`--limit`** — Bestimmt die maximale Zahl der Übereinstimmungen innerhalb eines bestimmten Zeitraums im Format ***<value>/<period>***. Mit **`--limit 5/hour`** darf die Regel beispielsweise nur **5** Mal pro Stunde übereinstimmen.

    Die Zeiträume können in Sekunden, Minuten, Stunden oder Tagen angegeben werden.

    Wenn keine Angaben zur Anzahl oder Zeit gemacht werden, wird der Standardwert **`3/hour`** angenommen.

  - **`--limit-burst`** — Setzt eine Grenze für die Anzahl von Paketen, die gleichzeitig mit einer Regel übereinstimmen können.

    Diese Option wird als ganzzahliger Wert angegeben und sollte zusammen mit der Option **`--limit`** verwendet werden.

    Wird kein Wert angegeben, so wird der Standardwert fünf (5) angenommen.

- **`state`**-Modul — Ermöglicht Übereinstimmungen nach Zustand.

  Das **`state`**-Modul akzeptiert die folgenden Optionen:

  - **`--state`** — Übereinstimmung mit einem Paket, das folgenden Verbindungszustand hat:

    - **`ESTABLISHED`** — Das übereinstimmende Paket gehört zu anderen Paketen in einer bestehenden Verbindung. Sie müssen diesen Zustand akzeptieren, wenn Sie eine Verbindung zwischen Client und Server aufrecht erhalten möchten.
    - **`INVALID`** — Das übereinstimmende Paket kann nicht mit einer bekannten Verbindung verknüpft werden.
    - **`NEW`** — Das übereinstimmende Paket stellt entweder eine neue Verbindung her oder ist Teil einer Zwei-Wege-Verbindung, die vorher nicht gesehen wurde. Sie müssen diesen Zustand akzeptieren, wenn Sie neue Verbindungen zu einem Dienst erlauben möchten.
    - **`RELATED`** — Ein übereinstimmendes Paket stellt eine neue Verbindung her, die auf irgendeine Weise mit einer bestehenden Verbindung zusammenhängt. Ein Beispiel hierfür ist FTP, das eine Verbindung zur Kontrolle des Datenverkehrs (Port 21) und eine separate Verbindung zur Übertragung von Daten (Port 20) verwendet.

    Diese Verbindungszustände können auch in Kombination verwendet werden, wobei sie durch Kommas getrennt werden, wie z. B. **`-m state --state INVALID,NEW`**.

- **`mac`**-Modul — Ermöglicht Übereinstimmung anhand Hardware-MAC-Adressen.

  Das **`mac`**-Modul akzeptiert die folgende Option:

  - **`--mac-source`** — Überprüft die MAC-Adresse der Netzwerkkarte, die das Paket gesendet hat. Um eine MAC-Adresse von einer Regel auszuschließen, fügen Sie nach der **`--mac-source`**-Übereinstimmungsoption ein Ausrufezeichen (**`!`**) hinzu.

Werfen Sie einen Blick auf die Handbuchseite von **`iptables`** für weitere Übereinstimmungsoptionen, die über Module verfügbar sind.

2.6.2.5. Zieloptionen

Sobald ein Paket mit einer bestimmten Regel übereinstimmt, kann die Regel das Paket an viele verschiedene Ziele senden, an denen dann die jeweiligen Aktionen durchgeführt werden. Jede Kette hat ein Standardziel, das verwendet wird, wenn ein Paket keiner Regel entspricht oder wenn in den Regeln, mit dem das Paket übereinstimmt, kein Ziel angegeben ist.

Dies sind die Standardziele:

- ***<user-defined-chain>*** — Eine benutzerdefinierte Kette innerhalb der Tabelle. Namen von benutzerdefinierten Ketten müssen eindeutig sein. Dieses Ziel leitet das Paket an die angegebene Kette weiter.
- **ACCEPT** — Lässt das Paket zu dessen Bestimmungsort oder zu einer anderen Kette passieren.
- **DROP** — Das Paket wird ohne jegliche Antwort verworfen. Das System, das dieses Paket gesendet hat, wird nicht über das Verwerfen des Pakets benachrichtigt.
- **QUEUE** — Das Paket wird zur Warteschlange für die Bearbeitung durch eine Userspace-Applikation hinzugefügt.
- **RETURN** — Hält die Suche nach Übereinstimmungen des Pakets mit Regeln in der aktuellen Kette an. Wenn ein Paket mit einem **RETURN**-Ziel mit einer Regel in einer Kette übereinstimmt, die von einer anderen Kette aufgerufen wurde, wird das Paket an die erste Kette zurückgesendet, damit die Überprüfung wieder dort aufgenommen werden kann, wo sie unterbrochen wurde. Wenn die **RETURN**-Regel in einer integrierten Kette verwendet wird und das Paket nicht zu seiner vorherigen Kette zurückkehren kann, entscheidet das Standardziel für die aktuelle Kette, welche Maßnahme getroffen wird.

Zusätzlich sind Erweiterungen verfügbar, mithilfe derer verschiedene andere Ziele angegeben werden können. Diese Erweiterungen werden Zielmodule oder auch Übereinstimmungsoptionsmodule genannt, und die meisten treffen lediglich auf spezielle Tabellen und Situationen zu. Weitere Informationen zu Übereinstimmungsoptionsmodulen finden Sie unter Abschnitt 2.6.2.4.4, „Module mit zusätzlichen Übereinstimmungsoptionen“.

Es gibt viele erweiterte Zielmodule, von denen sich die meisten nur auf bestimmte Tabellen oder Situationen beziehen. Einige der gebräuchlichsten Zielmodule, die standardmäßig in Red Hat Enterprise Linux enthalten sind, werden nachfolgend aufgeführt:

- **LOG** — Protokolliert alle Pakete, die dieser Regel entsprechen. Da die Pakete vom Kernel protokolliert werden, bestimmt die Datei **/etc/syslog.conf**, wohin diese Protokolleinträge geschrieben werden. Standardmäßig werden sie in der Datei **/var/log/messages** abgelegt.

  Nach dem **LOG**-Ziel können verschiedene, zusätzliche Optionen verwendet werden, um die Art der Protokollierung zu bestimmen:

  - **--log-level** — Bestimmt die Prioritätsstufe eines Protokollereignisses. Auf den Handbuchseiten von **syslog.conf** finden Sie eine Liste der Prioritätsstufen.
  - **--log-ip-options** — Protokolliert alle im Header eines IP-Pakets enthaltenen Optionen.
  - **--log-prefix** — Fügt beim Schreiben einer Protokollzeile eine Zeichenkette von bis zu 29 Zeichen vor der Protokollzeile ein. Dies ist beim Schreiben von syslog-Filtern im Zusammenhang mit der Paketprotokollierung sehr nützlich.

    Anmerkung

    Aufgrund eines Problems mit dieser Option sollten sie ein Leerzeichen hinter dem ***log-prefix***-Wert einfügen.

- **`--log-tcp-options`** — Protokolliert alle im Header eines TCP-Pakets enthaltenen Optionen.
- **`--log-tcp-sequence`** — Schreibt die TCP-Sequenznummer für das Paket in die Protokolldatei.

- **`REJECT`** — Sendet ein Fehlerpaket an das entfernte System zurück und verwirft das Paket.

  Mit dem **`REJECT`**-Ziel kann die **`--reject-with`** ***<type>***-Option verwendet werden (wobei ***<type>*** die Art der Zurückweisung angibt), um mehr Details zusammen mit dem Fehlerpaket zu senden. Die Meldung **`port-unreachable`** ist die standardmäßige Fehlermeldung, wenn keine andere Option angewandt wurde. Eine vollständige Liste der verfügbaren ***<type>***-Optionen finden Sie auf der **`iptables`**-Handbuchseite.

Andere Zielerweiterungen, die für das IP-Masquerading unter Verwendung der **`nat`**-Tabelle oder für Paketänderung mithilfe der **`mangle`**-Tabelle nützlich sind, finden Sie auf der **`iptables`**-Handbuchseite.

2.6.2.6. Auflistungsoptionen

Der standardmäßige Befehl zum Auflisten, **`iptables -L [<chain-name>]`**, bietet eine sehr allgemeine Übersicht über die aktuellen Ketten der standardmäßigen Filtertabelle. Doch es gibt zusätzliche Optionen, die weitere Informationen liefern:

- **`-v`** — Zeigt eine ausführliche Ausgabe an, wie z. B. die Anzahl der Pakete und Bytes, die jede Kette abgearbeitet hat, die Anzahl der Pakete und Bytes, die von jeder Regel auf Übereinstimmung überprüft wurden und welche Schnittstellen für eine bestimmte Regel zutreffen.
- **`-x`** — Erweitert die Zahlen auf ihre exakten Werte. In einem ausgelasteten System kann die Anzahl der Pakte und Bytes, die von einer bestimmten Kette oder Regel verarbeitet werden, auf **`Kilobytes`**, **`Megabytes`** und **`Gigabytes`** abgekürzt werden. Diese Option erzwingt die Anzeige der vollständigen Zahl.
- **`-n`** — Zeigt IP-Adressen und Portnummern in numerischem Format an, statt im standardmäßigen Hostnamen- und Netzwerkdienst-Format.
- **`--line-numbers`** — Listet Regeln in jeder Kette neben deren numerischer Position in der Kette auf. Diese Option ist nützlich, wenn Sie versuchen, eine bestimmte Regel aus einer Kette zu entfernen oder zu bestimmen, wo eine Regel in einer Kette eingefügt werden soll.
- **`-t <table-name>`** — Gibt einen Tabellennamen an. Falls nicht angegeben, wird standardmäßig die Filtertabelle verwendet.

2.6.3. Speichern von IPTables-Regeln

Regeln, die mit dem **`iptables`**-Befehl erstellt wurden, werden zunächst nur im Arbeitsspeicher bewahrt. Wird das System neu gestartet, bevor die **`iptables`**-Regeln gespeichert wurden, gehen diese Regeln verloren. Wenn Sie möchten, dass Netfilter-Regeln dauerhaft wirksam sind, müssen sie abgespeichert werden. Führen Sie dazu folgenden Befehl als Root aus:

```
/sbin/service iptables save
```

Dadurch wird das **`iptables`**-init-Skript angewiesen, das **`/sbin/iptables-save`**-Programm auszuführen und die aktuelle **`iptables`**-Konfiguration in die **`/etc/sysconfig/iptables`**-Datei zu schreiben. Die bestehende **`/etc/sysconfig/iptables`**-Datei wird unter **`/etc/sysconfig/iptables.save`** gespeichert.

Beim nächsten Systemstart wendet das **`iptables`**-init-Skript die in **`/etc/sysconfig/iptables`** gespeicherten Regeln mittels des **`/sbin/iptables-restore`**-Befehls erneut an.

Es ist grundsätzlich empfehlenswert, eine neue **`iptables`**-Regel immer erst zu testen, bevor sie in der **`/etc/sysconfig/iptables`**-Datei abgespeichert wird. Sie können die **`iptables`**-Regeln aber auch

von einer Datei eines anderen Systems in diese Datei kopieren, wodurch sie in kurzer Zeit ganze Gruppen von **iptables**-Regeln an verschiedene Rechner verteilen können.

Weiterhin haben Sie die Möglichkeit, die iptables-Regeln in einer separaten Datei zur weiteren Verteilung, zur Sicherung oder anderen Zwecken zu speichern. Um Ihre IPTables-Regeln zu speichern, geben Sie als Root folgenden Befehl ein:

```
[root@myServer ~]# iptables-save > <filename> wobei <filename> ein
benutzerdefinierter Name für Ihr Regelset ist.
```

Wichtig

Wenn Sie die **/etc/sysconfig/iptables**-Datei an andere Rechner verteilen, müssen Sie **/sbin/service iptables restart** ausführen, damit die neuen Regeln wirksam werden.

Anmerkung

Beachten Sie bitte den Unterschied zwischen dem **iptables**-*Befehl* (**/sbin/iptables**), der dazu verwendet wird, die Tabellen und Ketten zu handhaben, die die **iptables**-Funktionalität darstellen, und dem **iptables**-*Dienst* (**/sbin/iptables service**), der zum Aktivieren und Deaktivieren des **iptables**-Dienstes selbst verwendet wird.

2.6.4. IPTables Kontrollskripte

In Red Hat Enterprise Linux gibt es zwei grundlegende Methoden zur Steuerung von **iptables**:

- **Firewall-Konfigurationstool** (**system-config-selinux**) — Eine grafische Benutzeroberfläche zum Erstellen, Aktivieren und Speichern von einfachen Firewall-Regeln. Weitere Informationen über die Verwendung dieses Tools finden Sie unter Abschnitt 2.5.2, „Grundlegende Firewall-Konfiguration“.
- **/sbin/service iptables *<option>*** — Wird verwendet, um verschiedene Funktionen von **iptables** zu handhaben, unter Verwendung des init-Skripts von IPTables. Die folgenden Optionen stehen zur Verfügung:
  - **start** — Ist eine Firewall konfiguriert (d. h. **/etc/sysconfig/iptables** ist vorhanden), werden alle laufenden **iptables** komplett beendet und dann mit dem Befehl **/sbin/iptables-restore** gestartet. Diese Option funktioniert nur dann, wenn das **ipchains** Kernel-Modul nicht geladen ist. Um zu überprüfen, ob dieses Modul geladen ist, führen Sie als Root folgenden Befehl aus:

    ```
    [root@MyServer ~]# lsmod | grep ipchains
    ```

    Wenn dieser Befehl keine Ausgabe zurückgibt, ist das Modul nicht geladen. Falls nötig, können Sie das Modul mit dem Befehl **/sbin/rmmod** entfernen.
  - **stop** — Falls eine Firewall ausgeführt wird, werden die Firewall-Regeln im Speicher gelöscht und alle IPTables-Module und Helfer entladen.

    Wenn die **IPTABLES_SAVE_ON_STOP**-Direktive in der Konfigurationsdatei **/etc/sysconfig/iptables-config** vom Standardwert auf **yes** geändert wurde, werden die augenblicklichen Regeln unter **/etc/sysconfig/iptables** gespeichert und jede bestehende Regel nach **/etc/sysconfig/iptables.save** verschoben.

Werfen Sie einen Blick auf Abschnitt 2.6.4.1, „Konfigurationsdatei der IPTables-Kontrollskripte“ für weitere Informationen zur **iptables-config**-Datei.

- **restart** — Falls eine Firewall ausgeführt wird, werden die Firewall-Regeln im Speicher gelöscht und die Firewall, sollte sie in **/etc/sysconfig/iptables** konfiguriert sein, neu gestartet. Diese Option funktioniert nur dann, wenn das **ipchains** Kernel-Modul nicht geladen ist.

  Wenn die **IPTABLES_SAVE_ON_RESTART**-Direktive der Konfigurationsdatei **/etc/sysconfig/iptables-config** vom Standardwert auf **yes** geändert wurde, werden die augenblicklichen Regeln unter **/etc/sysconfig/iptables** gespeichert und jede bestehende Regel nach **/etc/sysconfig/iptables.save** verschoben.

  Werfen Sie einen Blick auf Abschnitt 2.6.4.1, „Konfigurationsdatei der IPTables-Kontrollskripte“ für weitere Informationen zur **iptables-config**-Datei.

- **status** — Zeigt den Status einer Firewall an und listet alle aktiven Regeln auf.

  Die Standardkonfiguration für diese Option zeigt die IP-Adressen in jeder Regel an. Um Informationen über Domain- und Hostnamen anzuzeigen, bearbeiten Sie die Datei **/etc/sysconfig/iptables-config** und setzen den Wert von **IPTABLES_STATUS_NUMERIC** auf **no**. Werfen Sie einen Blick auf Abschnitt 2.6.4.1, „Konfigurationsdatei der IPTables-Kontrollskripte“ für weitere Informationen zur **iptables-config**-Datei.

- **panic** — Löscht alle Firewall-Regeln. Die Richtlinie aller konfigurierten Tabellen wird auf **DROP** gesetzt.

  Diese Option kann nützlich sein, wenn ein Server im Verdacht steht, kompromittiert worden zu sein. Anstatt das System physisch vom Netzwerk zu trennen oder es herunterzufahren, können Sie mithilfe dieser Option jeglichen weiteren Netzwerkverkehr stoppen und gleichzeitig den Rechner in einem Zustand belassen, der eine Analyse oder andere forensische Untersuchungen ermöglicht.

- **save** — Speichert Firewall-Regeln mittels **iptables-save** nach **/etc/sysconfig/iptables**. Werfen Sie einen Blick auf Abschnitt 2.6.3, „Speichern von IPTables-Regeln“ für weitere Informationen.

Anmerkung

Um die gleichen Initskript-Befehle zu verwenden, um Netfilter für IPv6 zu steuern, ersetzen Sie **iptables** durch **ip6tables** in den in diesem Abschnitt angegebenen **/sbin/service**-Befehlen. Für weitere Informationen zu IPv6 und Netfilter, werfen Sie einen Blick auf Abschnitt 2.6.5, „IPTables und IPv6“.

2.6.4.1. Konfigurationsdatei der IPTables-Kontrollskripte

Das Verhalten des **iptables**-init-Skripts wird durch die Konfigurationsdatei **/etc/sysconfig/iptables-config** gesteuert. Nachfolgend sehen Sie eine Liste der in dieser Datei enthaltenen Direktiven:

- **IPTABLES_MODULES** — Gibt eine durch Leerzeichen getrennte Liste von zusätzlichen **iptables**-Modulen an, die beim Aktivieren einer Firewall geladen werden, wie z. B. Verbindungs-Tracker und NAT-Helfer.
- **IPTABLES_MODULES_UNLOAD** — Entlädt Module beim Neustarten und Stoppen. Diese Direktive akzeptiert die folgenden Werte:
  - **yes** — Der Standardwert. Diese Option muss gesetzt sein, um einen richtigen Status für einen Firewall-Neustart oder -Stopp zu erhalten.

  - **no** — Diese Option sollte nur dann gesetzt sein, wenn es Probleme beim Entladen der Netfilter-Module gibt.
- **IPTABLES_SAVE_ON_STOP** — Speichert die aktuellen Firewall-Regeln unter **/etc/sysconfig/iptables**, wenn die Firewall gestoppt wird. Diese Direktive akzeptiert folgende Werte:
  - **yes** — Speichert vorhandene Regeln unter **/etc/sysconfig/iptables**, wenn die Firewall gestoppt wird. Die vorherige Version wird unter **/etc/sysconfig/iptables.save** abgelegt.
  - **no** — Der Standardwert. Bestehende Regeln werden nicht gespeichert, wenn die Firewall gestoppt wird.
- **IPTABLES_SAVE_ON_RESTART** — Speichert die aktuellen Firewall-Regeln, wenn die Firewall neu gestartet wird. Diese Direktive akzeptiert die folgenden Werte:
  - **yes** — Speichert bestehende Regeln unter **/etc/sysconfig/iptables**, wenn die Firewall neu gestartet wird. Die vorherige Version wird dabei unter **/etc/sysconfig/iptables.save** abgelegt.
  - **no** — Der Standardwert. Bestehende Regeln werden nicht gespeichert, wenn die Firewall neu gestartet wird.
- **IPTABLES_SAVE_COUNTER** — Speichert und stellt alle Paket- und Byte-Zähler in allen Ketten und Regeln wieder her. Diese Direktive akzeptiert die folgenden Werte:
  - **yes** — Speichert die Werte der Zähler.
  - **no** — Der Standardwert. Speichert die Werte der Zähler nicht.
- **IPTABLES_STATUS_NUMERIC** — Gibt die IP-Adressen in numerischer Form aus anstelle der Domain- oder Hostnamen. Diese Direktive akzeptiert die folgenden Werte:
  - **yes** — Der Standardwert. Gibt lediglich IP-Adressen in der Statusanzeige aus.
  - **no** — Gibt Domain- oder Hostnamen in der Statusanzeige aus.

2.6.5. IPTables und IPv6

Wenn das Paket **iptables-ipv6** installiert ist, kann der Netfilter in Red Hat Enterprise Linux das neueste IPv6-Internetprotokoll filtern. Der Befehl zur Verwaltung des IPv6-Netfilters lautet **ip6tables**.

Die meisten Direktiven für diesen Befehl sind identisch mit denen von **iptables**, mit der Ausnahme, dass die **nat**-Tabelle noch nicht unterstützt wird. Infolgedessen ist es noch nicht möglich, IPv6 Network-Address-Translation-Aufgaben, wie z. B. Masquerading oder Port-Forwarding, durchzuführen.

Regeln für **ip6tables** werden in der Datei **/etc/sysconfig/ip6tables** gespeichert. Vorherige, durch die **ip6tables**-init-Skripte gespeicherte Regeln werden in der Datei **/etc/sysconfig/ip6tables.save** abgelegt.

Die Konfigurationsoptionen für **ip6tables**-init-Skripte werden in **/etc/sysconfig/ip6tables-config** gespeichert und die Namen der jeweiligen Direktiven unterscheiden sich leicht von ihren **iptables**-Gegenstücken.

Das Äquivalent zur **iptables-config**-Direktive **IPTABLES_MODULES** ist zum Beispiel **IP6TABLES_MODULES** in der **ip6tables-config**-Datei.

2.6.6. Zusätzliche Informationsquellen

In den nachfolgend aufgeführten Quellen finden Sie zusätzliche Informationen zur Paketfilterung mit **iptables**.

- Abschnitt 2.5, „Firewalls“ — Enthält ein Kapitel über die Rolle von Firewalls innerhalb einer

umfassenden Sicherheitsstrategie, sowie Strategien zum Erstellen von Firewall-Regeln.

2.6.6.1. Installierte IPTables-Dokumentation

- **`man iptables`** — Enthält eine Beschreibung von **`iptables`**, sowie eine umfangreiche Liste verschiedener Ziele, Optionen und Übereinstimmungserweiterungen.

2.6.6.2. Hilfreiche IPTables-Websites

- http://www.netfilter.org/ — Die Homepage des Netfilter/iptables-Projekts. Enthält ausgewählte Informationen zu **`iptables`** sowie ein FAQ zu spezifischen Problemen und verschiedene hilfreiche Handbücher von Rusty Russell, dem Linux-IP-Firewall-Maintainer. In diesen Anleitungen werden Themen, wie z. B. grundlegende Netzwerkkonzepte, Kernel-Paketfilterung und NAT-Konfigurationen behandelt.
- http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html — Eine Einführung in die Art und Weise, wie sich Pakete durch den Linux-Kernel bewegen, sowie eine Übersicht zur Erstellung von einfachen **`iptables`**-Befehlen.

---

[11] Da sich das System-BIOS von Hersteller zu Hersteller unterscheidet, unterstützen u. U. einige nicht den Passwortschutz beider Typen, während andere vielleicht nur einen Typ und nicht den anderen unterstützen.

[12] GRUB akzeptiert auch Klartextpasswörter, aus Sicherheitsgründen wird jedoch empfohlen, die md5-Hash-Version zu verwenden.

[13] Dieser Zugang unterliegt nach wie vor den von SELinux verhängten Einschränkungen, falls aktiviert.

Kapitel 3. Verschlüsselung

Grundsätzlich gibt es zwei Arten von Daten, die geschützt werden müssen: Data-at-rest (ruhende, gespeicherte Daten) und Data-in-motion (Daten während der Übertragung). Diese verschiedenen Arten von Daten werden auf ähnliche Weise unter Verwendung ähnlicher Technologien gesichert, die Implementierungen können sich jedoch maßgeblich unterscheiden. Keine einzelne Sicherheitsimplementierung kann alle möglichen Arten von Bedrohungen verhindern, da dieselben Daten zu einem bestimmten Zeitpunkt ruhend sind, sich an einem anderen Zeitpunkt jedoch in Übertragung befinden können.

3.1. Ruhende Daten

Ruhende Daten (auch "Data-at-rest" genannt) sind Daten, die auf einer Festplatte, CD, DVD, Magnetband oder einem anderen Datenträger gespeichert sind. Die größte Gefahr für diese Daten besteht in einem simplen physischen Diebstahl. Laptops an Flughäfen, CDs in der Post und verlegte Backup-Bänder sind Beispiele dafür, wie Daten durch Diebstahl in die falschen Hände geraten können. Falls die Daten auf dem Datenträger jedoch verschlüsselt waren, brauchen Sie sich wenigstens weniger Sorgen darum zu machen, dass die Daten ausgelesen und böswillig missbraucht werden.

3.2. Vollständige Festplattenverschlüsselung

Die vollständige Verschlüsselung der Festplatte oder Partition ist der beste Weg, um Ihre Daten zu schützen. Nicht nur jede Datei ist geschützt, sondern auch der temporäre Speicher, der unter Umständen Teile dieser Dateien enthält. Eine vollständige Festplattenverschlüsselung schützt Ihre gesamten Dateien, Sie brauchen also nicht auszuwählen, welche Dateien geschützt werden sollen und riskieren daher auch nicht, wichtige Dateien zu vergessen.

Red Hat Enterprise Linux 6 beinhaltet native Unterstützung für LUKS-Verschlüsselung. LUKS verschlüsselt Ihre Festplattenpartitionen, so dass Ihre Daten geschützt sind, während Ihr Computer ausgeschaltet ist. Es schützt Ihre Daten auch vor Angreifern, die sich mittels Einzelbenutzermodus an Ihrem Computer anzumelden versuchen oder anderweitig darauf zuzugreifen versuchen.

Lösungen zur vollständigen Festplattenverschlüsselung, wie z. B. LUKS, schützen die Daten nur, während der Computer ausgeschaltet ist. Sobald der Computer eingeschaltet ist und LUKS die Festplatten entschlüsselt hat, sind die Dateien nun für jeden zugänglich, der auch normalerweise Zugriff auf diese Dateien hat. Um Ihre Dateien zu schützen, während Ihr Computer eingeschaltet ist, verwenden Sie die vollständige Festplattenverschlüsselung zusammen mit einer anderen Lösung wie z. B. einer dateibasierten Verschlüsselung. Denken Sie zudem daran, Ihren Computer zu sperren, sobald Sie sich davon entfernen. Ein passwortgeschützter Bildschirmschoner, der nach einigen Minuten der Inaktivität automatisch startet, ist eine gute Möglichkeit, Eindringlinge abzuhalten.

3.3. Dateibasierte Verschlüsselung

GnuPG (GPG) ist eine quelloffene Version von PGP, die es Ihnen ermöglicht, eine Datei oder eine E-Mail zu signieren und/oder zu verschlüsseln. Dies bewahrt die Integrität der Nachricht oder Datei und schützt vertrauliche Informationen innerhalb der Nachricht oder Datei. Im Falle von E-Mail bietet GPG doppelten Schutz. Es schützt nicht nur die ruhenden Daten, sondern auch Daten während der Übertragung, sobald die Nachricht über das Netzwerk gesendet wird.

Dateibasierte Verschlüsselung soll eine Datei schützen, nachdem Sie Ihren Computer verlassen hat, z. B. wenn Sie eine CD per Post verschicken. Einige Lösungen für dateibasierte Verschlüsselung lassen Spuren der verschlüsselten Dateien zurück, die ein Angreifer, der Zugriff auf Ihren Computer erlangt, unter Umständen zur Wiederherstellung der ursprünglichen Datei nutzen kann. Um die Inhalte dieser

Dateien vor Angreifern zu schützen, die Zugriff auf Ihren Computer erlangen, verwenden Sie dateibasierte Verschlüsselung zusammen mit einer anderen Lösung wie z. B. der vollständigen Festplattenverschlüsselung.

3.4. Daten in Übertragung

Daten in Übertragung (auch "Data-in-Motion" genannt) sind Daten, die über ein Netzwerk übertragen werden. Die größten Gefahren für Daten in Übertragung sind das Abfangen oder Ändern dieser Daten. Ihr Benutzername und Passwort sollten niemals schutzlos über ein Netzwerk gesendet werden, da diese abgefangen und von Dritten verwendet werden können, um Zugriff auf sensible Daten zu erlangen. Andere private Daten wie z. B. Kontoinformationen sollten bei der Übertragung über ein Netzwerk ebenfalls geschützt werden. Falls die Netzwerksitzung verschlüsselt ist, brauchen Sie sich dagegen weniger Sorgen darum zu machen, dass die Daten bei der Übertragung abgefangen oder verändert werden.

Daten in Übertragung sind deshalb besonders gefährdet, da sich der Angreifer nicht tatsächlich in der Nähe des Computers befinden muss, auf dem die Daten gespeichert werden, sondern nur irgendwo auf dem Übertragungsweg. Verschlüsselungstunnel können die Daten auf Ihrem Kommunikationsweg schützen.

3.5. Virtuelle Private Netzwerke

Virtuelle Private Netzwerke (Virtual Private Networks oder kurz VPNs) bieten verschlüsselte Tunnel zwischen Computern oder Netzwerken von Computern über alle Ports hinweg. Ist ein VPN eingerichtet, wird sämtlicher Netzwerkverkehr vom Client durch den verschlüsselten Tunnel zum Server weitergeleitet. Das bedeutet, dass sich der Client logisch auf demselben Netzwerk wie der Server befindet, mit dem er über das VPN verbunden ist. VPNs sind weit verbreitet sowie einfach einzurichten und zu handhaben.

3.6. Secure Shell

Secure Shell (SSH) ist ein leistungsstarkes Netzwerkprotokoll, das zur Kommunikation mit einem anderen System über einen sicheren Tunnel verwendet wird. Die mit SSH übertragenen Daten sind verschlüsselt und vor dem Abfangen durch Angreifer geschützt. Zudem kann eine kryptografische Anmeldung verwendet werden, um eine bessere Authentifizierungsmethode als herkömmliche Benutzernamen und Passwörter zu bieten.

SSH ist sehr einfach zu aktivieren. Sobald der sshd-Dienst gestartet wird, akzeptiert das System Verbindungen und erlaubt Zugriff auf das System, wenn beim Verbindungsvorgang eine korrekte Benutzername/Passwort-Kombination angegeben wird. Der standardmäßige TCP-Port für den SSH-Dienst ist 22, dies kann jedoch geändert werden, indem Sie die Konfigurationsdatei */etc/ssh/sshd_config* bearbeiten und den Dienst neu starten. Diese Datei enthält zudem weitere Konfigurationsoptionen für SSH.

Secure Shell (SSH) bietet zudem verschlüsselte Tunnel zwischen Computern, nutzt jedoch nur einen einzigen Port. Port-Weiterleitung kann über einen SSH-Tunnel erfolgen und die Daten werden bei der Übertragung über diesen Tunnel verschlüsselt, allerdings ist die Verwendung von Port-Weiterleitung nicht so flüssig wie ein VPN.

3.7. OpenSSL PadLock Engine

Die PadLock Engine steht auf einigen VIA C3 Prozessoren (Nehemia) zur Verfügung und ermöglicht extrem schnelle Hardware-Verschlüsselung und -Entschlüsselung.

Anmerkung

64-bit Systeme enthalten keine Unterstützung für VIA Padlock.

Um es zu aktivieren, bearbeiten Sie die Datei **/etc/pki/tls/openssl.cnf** und fügen Folgendes am Anfang der Datei hinzu:

```
openssl_conf = openssl_init
```

Fügen Sie anschließend Folgendes am Ende der Datei hinzu:

```
[openssl_init]
engines = openssl_engines

[openssl_engines]
padlock = padlock_engine

[padlock_engine]
default_algorithms = ALL
dynamic_path = /usr/lib/openssl/engines/libpadlock.so
init = 1
```

Führen Sie den folgenden Befehl aus um zu überprüfen, ob das Modul aktiviert ist:

```
# openssl engine -c -tt
```

Führen Sie den folgenden Befehl aus, um die Geschwindigkeit zu überprüfen:

```
# openssl speed aes-128-cbc
```

Führen Sie einen Befehl wie den folgenden aus, um die Geschwindigkeit von OpenSSH zu überprüfen:

```
# dd if=/dev/zero count=100 bs=1M | ssh -c aes128-cbc
localhost "cat >/dev/null"
```

Mehr Informationen über die VIA PadLock Engine finden Sie unter den folgenden URLs:
http://www.logix.cz/michal/devel/padlock/ und http://www.via.com.tw/en/initiatives/padlock/.

3.8. LUKS-Festplattenverschlüsselung

Linux Unified Key Setup-on-disk-format (kurz LUKS) ermöglicht Ihnen die Verschlüsselung von Partitionen auf Ihrem Linux-Rechner. Dies ist besonders wichtig bei mobilen Rechnern und Wechseldatenträgern. LUKS erlaubt multiple Benutzerschlüssel zur Verschlüsselung eines Master-Schlüssels, der zur gesamten Verschlüsselung der Partition genutzt wird.

3.8.1. LUKS-Implementierung in Red Hat Enterprise Linux

Red Hat Enterprise Linux 6 setzt LUKS zur Dateisystemverschlüsselung ein. Standardmäßig ist die Option zur Verschlüsselung der Dateisysteme bei der Installation nicht ausgewählt. Falls Sie die Option zur Festplattenverschlüsselung auswählen, werden Sie zur Eingabe einer Passphrase aufgefordert, die dann bei jedem Hochfahren Ihres Rechners abgefragt wird. Diese Passphrase entschlüsselt dann den

Schlüssel, der zur gesamten Verschlüsselung der Partition verwendet wurde. Falls Sie die standardmäßige Partitionstabelle anpassen, können Sie wählen, welche Partitionen Sie verschlüsseln möchten. Dies wird in den Partitionstabelleneinstellungen festgelegt.

Der standardmäßige Schlüssel für LUKS (siehe **`cryptsetup --help`**) ist aes-cbc-essiv:sha256 (ESSIV - Encrypted Salt-Sector Initialization Vector). Beachten Sie, dass das Installationsprogramm **Anaconda** standardmäßig den XTS-Modus (aes-xts-plain64) verwendet. Die standardmäßige Schlüsselgröße für LUKS ist 256 Bits. Die standardmäßige Schlüsselgröße für LUKS mit **Anaconda** (XTS-Modus) ist 512 Bits. Verfügbare Schlüssel sind:

- AES - Advanced Encryption Standard - FIPS PUB 197
- Twofish (Eine 128-Bit Blockchiffre)
- Serpent
- cast5 - RFC 2144
- cast6 - RFC 2612

3.8.2. Manuelle Verschlüsselung von Verzeichnissen

Warnung

Die folgenden Schritte löschen sämtliche Daten auf der zu verschlüsselnden Partition. Sie werden sämtliche Daten verlieren! Erstellen Sie eine Sicherungskopie Ihrer Daten auf einem externen Speichergerät, bevor Sie mit diesen Schritten fortfahren.

3.8.3. Schrittweise Anleitung

1. Wechseln Sie zu Runlevel 1: **`telinit 1`**
2. Hängen Sie Ihre vorhandene /home-Partition aus: **`umount /home`**
3. Sollte dies fehlschlagen, verwenden Sie **`fuser`**, um noch aktive Prozesse auf /home zu finden und zu beenden: **`fuser -mvk /home`**
4. Vergewissern Sie sich, dass /home nicht mehr eingehängt ist: **`cat /proc/mounts | grep home`**
5. Füllen Sie Ihre Partition mit zufälligen Daten: **`dd if=/dev/urandom of=/dev/VG00/LV_home`** Dieser Vorgang kann mehrere Stunden dauern.

   Wichtig

   Dieser Vorgang ist von entscheidender Bedeutung für die Sicherung gegen Einbruchsversuche. Lassen Sie den Vorgang ggf. über Nacht laufen.

6. Initialisieren Sie Ihre Partition: **`cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`**
7. Öffnen Sie das neu verschlüsselte Gerät: **`cryptsetup luksOpen /dev/VG00/LV_home home`**
8. Vergewissern Sie sich, dass das Gerät vorhanden ist: **`ls -l /dev/mapper | grep home`**
9. Erstellen Sie ein Dateisystem: **`mkfs.ext3 /dev/mapper/home`**
10. Hängen Sie es ein: **`mount /dev/mapper/home /home`**
11. Überprüfen Sie, ob es sichtbar ist: **`df -h | grep home`**

12. Fügen Sie Folgendes zur /etc/crypttab hinzu: **`home /dev/VG00/LV_home none`**
13. Bearbeiten Sie Ihre /etc/fstab, entfernen Sie den alten Eintrag für /home und fügen Sie Folgendes hinzu: **`/dev/mapper/home /home ext3 defaults 1 2`**
14. Stellen Sie die standardmäßigen SELinux-Sicherheitskontexte wieder her: **`/sbin/restorecon -v -R /home`**
15. Starten Sie das Sytem neu: **`shutdown -r now`**
16. Der Eintrag in der /etc/crypttab-Datei weist Ihr System dazu an, beim Hochfahren zur Eingabe der **`luks`**-Passphrase aufzufordern.
17. Melden Sie sich als Root an und stellen Sie Ihre gesicherten Daten wieder her.

3.8.4. Ergebnis

Glückwunsch, Sie haben nun eine verschlüsselte Partition, auf der Ihre Daten sicher gespeichert sind, während der Rechner ausgeschaltet ist.

3.8.5. Hilfreiche Links

Für weiterführende Informationen über LUKS oder das Verschlüsseln von Festplatten unter Red Hat Enterprise Linux besuchen Sie bitte einen der folgenden Links:

- LUKS Homepage
- LUKS/cryptsetup FAQ
- LUKS - Linux Unified Key Setup
- HOWTO: Erstellen eines verschlüsselten physischen Datenträgers unter Verwendung einer zweiten Festplatte und pvmove

3.9. Verwenden von GNU Privacy Guard (GnuPG)

GPG wird zur Identifizierung Ihrer Person und zur Authentifizierung Ihrer Kommunikation eingesetzt, auch mit Personen, die Sie nicht kennen. GPG ermöglicht es jedem Leser einer GPG-signierten E-Mail, deren Authentizität zu überprüfen. Mit anderen Worten, GPG ermöglicht es Leuten, mit ziemlicher Sicherzeit zu bestätigen, dass von Ihnen signierte Nachrichten auch tatsächlich von Ihnen stammen. GPG ist hilfreich, da es Dritte daran hindert, Code zu ändern oder Nachrichtenwechsel abzufangen und deren Inhalt zu verändern.

3.9.1. Erstellen von GPG-Schlüsseln in GNOME

Installieren Sie das Seahorse-Dienstprogramm, das die Verwaltung von GPG-Schlüsseln erleichtert. Wählen Sie aus dem Hauptmenü **`System > Administration > Software hinzufügen/entfernen`** und warten Sie, bis die Applikation gestartet ist. Geben Sie **`Seahorse`** in das Textfeld ein und klicken Sie auf Suchen. Markieren Sie das Auswahlkästchen neben dem "seahorse"-Paket und klicken Sie auf "Anwenden", um die Software hinzuzufügen. Sie können **`Seahorse`** auch mithilfe des Befehls **`su -c "yum install seahorse"`** über die Befehlszeile installieren.

Um einen Schlüssel zu erstellen, wählen Sie aus dem Hauptmenü "Anwendungen > Zubehör" den Menüpunkt "Passwörter und Verschlüsselung", woraufhin die **`Seahorse`**-Applikation startet. Wählen Sie aus dem "Datei"-Menü "Neu", dann "PGP-Schlüssel". Klicken Sie anschließend auf "Weiter". Geben Sie Ihren vollständigen Namen, Ihre E-Mail-Adresse und einen optionalen Kommentar an (z. B.: John C. Smith, jsmith@example.com, The Man). Klicken Sie auf "Erstellen". Ein Dialogfeld erscheint, dass Sie zur Eingabe eines Passworts für den Schlüssel auffordert. Wählen Sie ein sicheres, jedoch einfach zu merkendes Passwort. Klicken Sie auf "OK", und der Schlüssel wird erstellt.

Warnung

Sollten Sie Ihr Passwort vergessen, kann der Schlüssel nicht mehr genutzt werden und sämtliche mit diesem Schlüssel verschlüsselten Daten sind verloren.

Um Ihre GPG-Schlüssel-ID zu finden, sehen Sie in der Spalte "Schlüssel-ID" neben dem neu erstellten Schlüssel nach. Wenn Sie nach der Schlüsel-ID gefragt werden, sollten Sie der Schlüssel-ID in den meisten Fällen "0x" voranstellen, also z. B. "0x6789ABCD". Sie sollten eine Sicherungskopie Ihres Schlüssels anlegen und diesen an einem sicheren Ort aufbewahren.

3.9.2. Erstellen von GPG-Schlüsseln in KDE

Starten Sie das KGpg-Programm aus dem Hauptmenü über Anwendungen > Dienstprogramme > Verschlüsselungs-Tool. Falls Sie KGpg noch nie zuvor benutzt haben, leitet Sie das Programm durch die nötigen Schritte zur Erstellung Ihres eigenen GPG-Schlüsselpaars. Ein Dialogfeld erscheint, das Sie zur Erstellung eines neuen Schlüsselpaares auffordert. Geben Sie Ihren Namen, Ihre E-Mail-Adresse und optional einen Kommentar ein. Sie können zudem ein Ablaufdatum für Ihren Schlüssel wählen, sowie die Stärke (Anzahl der Bits) und Algorithmen der Schlüssel. Das nächste Dialogfeld fordert Sie zur Eingabe Ihres Passworts auf. Daraufhin erscheint Ihr Schlüssel im Hauptfenster von **`KGpg`**.

Warnung

Sollten Sie Ihr Passwort vergessen, kann der Schlüssel nicht mehr genutzt werden und sämtliche mit diesem Schlüssel verschlüsselten Daten sind verloren.

Um Ihre GPG-Schlüssel-ID zu finden, sehen Sie in der Spalte "Schlüssel-ID" neben dem neu erstellten Schlüssel nach. Wenn Sie nach der Schlüsel-ID gefragt werden, sollten Sie der Schlüssel-ID in den meisten Fällen "0x" voranstellen, also z. B. "0x6789ABCD". Sie sollten eine Sicherungskopie Ihres Schlüssels anlegen und diesen an einem sicheren Ort aufbewahren.

3.9.3. Erstellen von GPG-Schlüsseln per Befehlszeile

Führen Sie den folgenden Shell-Befehl aus: **`gpg --gen-key`**

Dieser Befehl generiert ein Schlüsselpaar, das aus einem öffentlichen und einem privaten Schlüssel besteht. Andere Leute können Ihren öffentlichen Schlüssel nutzen, um Ihre Nachrichten zu authentifizieren und/oder zu entschlüsseln. Verbreiten Sie Ihren öffentlichen Schlüssel so weit wie möglich, insbesondere an Personen, von denen Sie wissen, dass sie authentifizierte Nachrichten von Ihnen erhalten wollen, z. B. eine Mailing-Liste.

Eine Reihe von Eingabeaufforderungen führt Sie durch den Vorgang. Drücken Sie die **`Eingabetaste`**, um einen Standardwert zuzuweisen, falls gewünscht. Die erste Eingabeaufforderung fordert Sie zur Auswahl auf, welche Art von Schlüssel Sie bevorzugen:

"Bitte wählen Sie, welche Art von Schlüssel Sie möchten: (1) DSA und ElGamal (voreingestellt) (2) DSA und Elgamal (3) DSA (nur unterschreiben/beglaubigen) (4) RSA (nur unterschreiben/beglaubigen) Ihre Auswahl?" In den meisten Fällen ist die Standardauswahl angemessen. Ein DSA/ElGamal-Schlüssel ermöglicht Ihnen nicht nur das Signieren von Nachrichten, sondern auch die Verschlüsselung von Dateien.

Wählen Sie als Nächstes die Schlüssellänge: Mindestschlüssellänge ist 768 Bits, Standardschlüssellänge ist 1024 Bits, und die höchste empfohlene Schlüssellänge ist 2048 Bits.

"Welche Schlüssellänge wünschen Sie? (1024)" Auch hier gilt, dass die Standardauswahl für die meisten Benutzer angemessen sein sollte und ein sehr hohes Maß an Sicherheit bietet.

Wählen Sie als Nächstes, wann die Gültigkeit des Schlüssels ablaufen soll. Es ist empfehlenswert, ein Ablaufdatum anzugeben, statt den Standardwert (kein Ablaufdatum) zu übernehmen. Falls beispielsweise die E-Mail-Adresse auf dem Schlüssel ungültig wird, kann ein Ablaufdatum andere Nutzer daran erinnern, diesen öffentlichen Schlüssel nicht länger zu verwenden.

"Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll. 0 = Schlüssel verfällt nie. d = Schlüssel verfällt nach n Tagen. w = Schlüssel verfällt nach n Wochen. m = Schlüssel verfällt nach n Monaten. y = Schlüssel verfällt nach n Jahren. Wie lange bleibt der Schlüssel gültig? (0)"

Wenn Sie beispielsweise den Wert **1y** eingeben, bleibt der Schlüssel für ein Jahr gültig. (Sie können dieses Ablaufdatum auch noch nach Erzeugung des Schlüssels ändern, sollten Sie es sich anders überlegen.)

Bevor das **gpg**-Programm die Signaturinformationen abfragt, erscheint die folgende Eingabeaufforderung: **Ist dies richtig (j/N)?** Tippen Sie **j**, um den Vorgang abzuschließen.

Geben Sie als Nächstes Ihren Namen und Ihre E-Mail-Adresse an. Denken Sie daran, dass der Sinn und Zweck dieser Schlüssel darin besteht, Sie als echte Person zu authentifizieren, geben Sie also Ihren richtigen Namen an. Verwenden Sie keine Aliasse oder Decknamen, da diese Ihre Identität verschleiern.

Geben Sie Ihre richtige E-Mail-Adresse für Ihren GPG-Schlüssel an. Falls Sie eine falsche oder erfundene E-Mail-Adresse angeben, erschweren Sie es anderen Leuten, Ihren öffentlichen Schlüssel zu finden. Dadurch wird die Authentifizierung Ihrer Nachrichten erschwert. Falls Sie diesen GPG-Schlüssel beispielsweise zur Vorstellung auf einer Mailing-Liste verwenden, geben Sie dieselbe E-Mail-Adresse an, die Sie auch für diese Mailing-Liste verwenden.

Verwenden Sie das Kommentarfeld, um Aliasse oder andere Informationen anzugeben. (Einige Leute nutzen unterschiedliche Schlüssel für unterschiedliche Zwecke und unterscheiden daher ihre Schlüssel anhand dieser Kommentare, wie z. B. "Büro" oder "Open Source Projekte".)

Geben Sie an der Eingabeaufforderung zur Bestätigung den Buchstaben "O" ein um fortzufahren, wenn alle Angeben korrekt sind, oder verwenden Sie eine der anderen Optionen, um etwaige Fehler zu beheben. Geben Sie abschließend ein Passwort für Ihren geheimen Schlüssel ein. Das **gpg**-Programm fordert Sie zur zweimaligen Eingabe des Passworts auf, um Tippfehler auszuschließen.

Das **gpg**-Programm generiert nun zufällige Daten, um Ihren Schlüssel so eindeutig wie möglich zu machen. Bewegen Sie während dieses Vorgangs Ihre Maus, tippen wahllos auf der Tastatur oder führen Sie andere Aufgaben auf dem System aus, um diesen Vorgang zu beschleunigen. Nachdem dieser Schritt abgeschlossen wurde, sind Ihre Schlüssel fertig und einsatzbereit:

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

Der Fingerabdruck des Schlüssels ist eine abgekürzte "Signatur" für Ihren Schlüssel. Anhand dessen können andere Personen überprüfen, ob sie Ihren öffenlichen Schlüssel unversehrt erhalten haben. Sie brauchen sich diesen Fingerabdruck nicht zu notieren. Sie können ihn jederzeit mit dem folgenden Befehl wieder anzeigen (ersetzen Sie dabei Ihre E-Mail-Adresse): **gpg --fingerprint jqdoe@example.com**

Ihre "GPG-Schlüssel-ID" besteht aus 8 Hexadezimalziffern, die den öffentlichen Schlüssel identifizieren. Im obigen Beispiel lautet die GPG-Schlüssel-ID 1B2AFA1C. Wenn Sie zur Angabe Ihrer Schlüssel-ID

aufgefordert werden, sollten Sie der Schlüssel-ID "0x" voranstellen, z. B. "0x1B2AFA1C".

Warnung

Sollten Sie Ihr Passwort vergessen, kann der Schlüssel nicht mehr genutzt werden und sämtliche mit diesem Schlüssel verschlüsselten Daten sind verloren.

3.9.4. Informationen zur asymmetrischen Verschlüsselung

1. Wikipedia - Public Key Cryptography
2. HowStuffWorks - Encryption

Kapitel 4. Allgemeine Prinzipien der Informationssicherheit

Die folgenden allgemeinen Prinzipien liefern einen Überblick über bewährte Sicherheitspraktiken:

- Verschlüsseln Sie alle Daten, die über das Netzwerk übertragen werden, um Man-in-the-Middle-Angriffe und das Abgreifen von Daten zu verhindern. Es ist insbesondere wichtig, sämtliche Daten zur Authentifikation wie z. B. Passwörter zu verschlüsseln.
- Installieren Sie nur die nötigste Software und führen Sie nur die nötigsten Dienste aus.
- Verwenden Sie Software und Tools zur Verbesserung der Sicherheit, wie z. B. Security-Enhanced Linux (SELinux) für Mandatory Access Control (MAC), Netfilter IPTables zur Paketfilterung (Firewall) und den GNU Privacy Guard (GnuPG) zur Verschlüsselung von Dateien.
- Falls möglich, führen Sie jeden Netzwerkdienst auf einem separaten System aus, um das Risiko zu verringern, dass ein kompromittierter Dienst zur Schädigung weiterer Dienste eingesetzt wird.
- Pflegen Sie die Benutzerkonten: Setzen Sie eine Richtlinie für sichere Passwörter durch und entfernen Sie ungenutzte Benutzerkonten.
- Sehen Sie regelmäßig die System- und Applikationsprotokolle durch. Standardmäßig werden Systemprotokolle rund um die Sicherheit in **`/var/log/secure`** und **`/var/log/audit/audit.log`** geschrieben. Beachten Sie auch, dass der Einsatz eines dedizierten Protokollservers es Angreifern erschwert, lokale Protokolle einfach zu ändern, um ihre Spuren zu verwischen.
- Melden Sie sich niemals als Root-Benutzer an, es sei denn, es ist absolut notwendig. Administratoren wird empfohlen, möglichst **`sudo`** zur Ausführung von Befehlen als Root zu nutzen. Benutzer, die **`sudo`** nutzen dürfen, können in der **`/etc/sudoers`**-Datei festgelegt werden. Verwenden Sie das **`visudo`**-Hilfsprogramm, um **`/etc/sudoers`** zu bearbeiten.

4.1. Tipps, Handbücher und Werkzeuge

Die US-amerikanische National Security Agency (NSA) bietet Leitfäden zur Abhärtung von Systemen sowie Tipps für viele verschiedene Betriebssysteme, um Regierungsbehören, Unternehmen und Privatleuten dabei zu helfen, Ihre Systeme vor möglichen Angriffen zu schützen. Die folgenden Handbücher bieten Leitfäden für Red Hat Enterprise Linux 6 (im PDF-Format):

- Hardening Tips for the Red Hat Enterprise Linux 5
- Guide to the Secure Configuration of Red Hat Enterprise Linux 5

Anmerkung

Wir verweisen an dieser Stelle auf das Handbuch zur Abhärtung von Red Hat Enterprise Linux 5 Systemen, bis ein entsprechendes Handbuch für Red Hat Enterprise Linux 6 veröffentlicht wird. Bis dahin beachten Sie bitte, dass das Handbuch zur Abhärtung von Red Hat Enterprise Linux 5 Systemen nur begrenzt auf Red Hat Enterprise Linux 6 übertragbar ist.

Die Defense Information Systems Agency (DISA) bietet Dokumentation, Checklisten und Tests, die Ihnen bei der Absicherung Ihres Systems helfen können (Information Assurance Support Environment). Der UNIX SECURITY TECHNICAL IMPLEMENTATION GUIDE (PDF) ist ein Handbuch speziell für UNIX-Sicherheit und setzt ein fortgeschrittenes Wissen über UNIX und Linux voraus.

Die DISA Unix Security Checklist liefert eine Sammlung von Dokumenten und Checklisten mit Informationen über richtige Besitzer und Modi für Systemdateien, bis hin zur Verwaltung von Patches.

Kapitel 5. Sichere Installation

Eine gute Sicherheitsstrategie beginnt bereits dann, wenn Sie die CD oder DVD zur Installation von Red Hat Enterprise Linux in Ihr Laufwerk einlegen. Wenn Sie Ihr System von Beginn an sicher konfigurieren, erleichtert Ihnen dies später das Implementieren zusätzlicher Sicherheitseinstellungen.

5.1. Festplattenpartitionen

Die US-amerikanische National Security Agency (NSA) empfiehlt, separate Partitionen anzulegen für /boot, /, /home, /tmp und /var/tmp. Die Gründe dafür sind jeweils unterschiedlich und werden im Folgenden für jede Partition einzeln erläutert.

/boot - Dies ist die erste Partition, die während des Starts vom System gelesen wird. Der Bootloader und die Kernel-Images, die zum Booten Ihres Systems in Red Hat Enterprise Linux genutzt werden, sind auf dieser Partition gespeichert. Diese Partition sollte nicht verschlüsselt werden. Falls diese Partition in / enthalten ist und diese Partition verschlüsselt wird oder anderweitig nicht verfügbar ist, wird Ihr System nicht booten können.

/home - Wenn Benutzerdaten (/home) in / gespeichert werden statt auf einer separaten Partition, kann die Partition voll werden und dadurch ein instabiles Betriebssystem verursachen. Auch ist es sehr viel einfacher, Ihr System auf eine neue Version von Red Hat Enterprise Linux zu aktualisieren, wenn Sie Ihre Daten in der /home Partition speichern, da diese bei der Installation nicht überschrieben wird. Falls die Root-Partition (/) beschädigt wird, könnten zudem Ihre Daten verloren gehen. Indem Sie jedoch eine separate Partition verwenden, ist das Risiko des Datenverlusts etwas geringer. Darüber hinaus können Sie auf diese Weise regelmäßige Backups dieser Partition durchführen.

/tmp und /var/tmp - Sowohl die /tmp als auch die /var/tmp Verzeichnisse werden für Daten genutzt, die nur für kürzere Zeit gespeichert werden müssen. Falls jedoch große Mengen an Daten diese Verzeichnisse überschwemmen, kann dies all Ihren Speicherplatz verbrauchen. Befinden sich diese Verzeichnisse unter / und tritt diese Situation auf, dann könnte Ihr System instabil werden und abstürzen. Aus diesem Grund ist es empfehlenswert, diese Verzeichnisse auf Ihre eigenen Partitionen zu legen.

5.2. Verwenden der LUKS-Partitionsverschlüsselung

Während des Installationsvorgangs wird Ihnen die Option zur Verschlüsselung Ihrer Partitionen geboten. Sie müssen eine Passphrase angeben, die den Verschlüsselungscode zur Sicherung der Daten auf dieser Partition aktiviert.

Kapitel 6. Software-Wartung

Die Software-Wartung ist von entscheidender Bedeutung, um ein System sicher zu halten. Es ist unerlässlich, Software-Patches umgehend nach deren Veröffentlichung anzuwenden, um Angreifer daran zu hindern, bekannte Sicherheitslücken zum Einstieg in Ihr System auszunutzen.

6.1. Installieren minimaler Software

Es wird im Allgemeinen empfohlen, nur benötigte Pakete zu installieren, da jede installierte Software auf Ihrem Computer potenziell eine Sicherheitslücke enthalten könnte. Wenn Sie von CD/DVD installieren, nutzen Sie die Möglichkeit, nur die gewünschten Pakete für die Installation auszuwählen. Sollten Sie später feststellen, dass Sie weitere Pakete benötigen, können Sie diese bei Bedarf später hinzufügen.

6.2. Planen und Konfigurieren von Sicherheitsaktualisierungen

Jede Software enthält Fehler. Oft können diese Fehler Sicherheitslücken verursachen, die Ihr System anfällig für böswillige Angreifer macht. Gewöhnlich werden solche Systeme Opfer von Angriffen, bei denen es versäumt wurde, verfügbare Patches anzuwenden. Sie sollten daher einen Plan haben, nach dem Sicherheits-Patches umgehend angewendet werden, um diese Sicherheitslücken zu schließen.

Auch private Benutzer sollten Sicherheitsaktualisierungen so bald wie möglich installieren. Sie können eine automatische Installation dieser Sicherheitsaktualisierungen konfigurieren, damit Sie nicht selbst daran denken müssen. Dies bringt allerdings ein gewisses Risiko mit sich, dass eine Aktualisierung einen Konflikt mit Ihrer Konfiguration oder mit anderer Software auf Ihrem System verursacht.

Fortgeschrittene private Benutzer sowie Systemadministratoren in Unternehmen sollten die Sicherheitsaktualisierungen zunächst testen und erst dann zur Installation einplanen. In der Zwischenzeit sollten weitere Sicherheitsmaßnahmen ergriffen werden, um das System zu schützen. Diese Sicherheitsmaßnahmen hängen von der jeweiligen Sicherheitslücke ab, könnten aber zum Beispiel aus zusätzlichen Firewall-Regeln, der Verwendung externer Firewalls oder Änderungen der Softwareeinstellungen bestehen.

6.3. Anpassen der automatischen Aktualisierungen

Red Hat Enterprise Linux ist standardmäßig dazu konfiguriert, täglich alle verfügbaren Aktualisierungen zu installieren. Falls Sie ändern möchten, wie Ihr System Aktualisierungen installiert, können Sie dies im Dialogfeld "Softwareaktualisierungs-Einstellungen" tun. Sie können dort den Zeitplan ändern, sowie die Art der anzuwendenden Aktualisierungen und Benachrichtigungen über verfügbare Aktualisierungen festlegen.

In Gnome finden Sie die Einstellungen für Aktualisierungen unter: **`System -> Einstellungen -> Software-Aktualisierungen`**. In KDE finden Sie dies unter: **`Anwendungen -> Einstellungen -> Software-Aktualisierungen`**.

6.4. Installieren signierter Pakete von bekannten Repositorys

Software-Pakete werden über Repositorys vertrieben. Alle bekannten Repositorys unterstützen das Signieren von Paketen. Die Paketsignatur nutzt die asymmetrische Kryptotechnologie um zu beweisen, dass das im Repository veröffentlichte Paket nicht verändert wurde, seit die Signatur darauf angewendet wurde. Dies bietet einen gewissen Schutz vor der Installation von Software, die mit böswilligen Absichten verändert wurde, nachdem das Paket erstellt wurde und bevor Sie das Paket heruntergeladen haben.

Wenn Sie zu viele verschiedene Repositorys verwenden, nicht vertrauenswürdige Repositorys oder

Repositorys mit unsignierten Paketen, so erhöht dies Ihr Risiko, dass bösartiger oder anfälliger Code in Ihr System eingeschleust wird. Wählen Sie daher mit Bedacht aus, welche Repositorys Sie zum Yum-/Software-Update hinzufügen.

Kapitel 7. Regierungsstandards und -reglementierungen

7.1. Einführung

Um ein gutes Maß an Sicherheit zu bewahren, kann Ihre Organisation sich nach den Sicherheitsspezifikationen, -standards und -reglementierungen von Regierung und Wirtschaft richten. Dieses Kapitel beschreibt einige dieser Standards und Reglementierungen.

7.2. Federal Information Processing Standard (FIPS)

Die Federal Information Processing Standard (FIPS) Publikaton 140-2 ist ein Standard zur Computersicherheit, entwickelt von einer Arbeitsgruppe bestehend aus Vertretern der U.S. Regierung und der Wirtschaft, um die Qualität von kryptografischen Modulen zu untersuchen. FIPS Publikationen (einschließlich 140-2) sind unter der folgenden URL erhältlich: http://csrc.nist.gov/publications/PubsFIPS.html. Beachten Sie, dass sich die Publikation 140-3 zum Zeitpunkt der Abfassung dieses Texts noch im Entwurfsstatus befindet und möglicherweise nicht den endgültigen Standard bildet. Der FIPS-Standard bietet vier Sicherheits-*Level*, um verschiedenen Branchen, Implementierungen kryptografischer Module und Unternehmensgrößen und -anforderungen gerecht zu werden. Diese Level werden nachfolgend beschrieben:

- Level 1 - Sicherheits-Level 1 bietet das geringste Level an Sicherheit. Nur einfache Sicherheitsanforderungen werden für ein kryptografisches Modul spezifiziert (z. B. muss mindestens ein bestätigter Algorithmus oder eine bestätigte Sicherheitsfunktion genutzt werden). Über diese einfachen Anforderungen hinaus sind in einem kryptografischen Modul des Sicherheits-Level 1 keine physischen Sicherheitsverfahren für Produktionskomponenten erforderlich. Ein Beispiel für ein kryptografisches Modul auf Sicherheits-Level 1 ist ein PC-Encryption-Board.
- Level 2 - Sicherheits-Level 2 verbessert die physischen Sicherheitsmaßnahmen eines kryptografischen Moduls des Sicherheits-Level 1 durch zusätzlich notwendige Originalitätssicherung mithilfe von Siegeln oder Beschichtungen oder mithilfe von Sicherheitsschlössern oder Abdeckungen für das Modul. Beschichtungen oder Siegel zur Originalitätssicherung werden auf einem kryptografischen Modul platziert, so dass die Beschichtung oder das Siegel gebrochen werden müssen, um physischen Zugriff auf die kryptografischen Schlüssel in Klartext und auf kritische Sicherheitsparameter (CSPs) innerhalb des Moduls zu erlangen. Sicherheitsschlösser oder Siegel zur Originalitätssicherung werden auf Abdeckungen oder Öffnungen platziert, um gegen unbefugten physischen Zugriff zu schützen.
- Level 3 - Zusätzlich zur den Sicherheitsmaßnahmen zur Originalitätssicherung für Sicherheits-Level 2 versucht das Sicherheits-Level 3 einen Angreifer daran zu hindern, Zugriff auf die kritischen Sicherheitsparameter innerhalb des kryptografischen Moduls zu erlangen. Die für Sicherheits-Level 3 nötigen physischen Sicherheitsmechanismen sollen mit hoher Wahrscheinlichkeit Versuche, auf das kryptografische Modul zuzugreifen, es zu verwenden oder zu verändern, erkennen und darauf reagieren. Zu den physischen Sicherheitsmechanismen gehören der Einsatz von stabilen Gehäusen sowie Schaltkreise zur Erkennung von Einbrüchen, die sämtliche kritische Sicherheitsparameter mit Nullen überschreiben, wenn die Abdeckungen oder Öffnungen des kryptografischen Moduls geöffnet werden.
- Level 4 - Das Sicherheits-Level 4 bietet das höchste Maß an Sicherheit, das in diesem Standard definiert ist. Bei diesem Sicherheits-Level bieten die physischen Sicherheitsmechanismen einen vollständigen Schutzschild um das kryptografische Modul, um alle unerlaubten physischen Zugriffe darauf zu erkennen und entsprechend zu reagieren. Ein wie auch immer geartetes Eindringen in das Gehäuse des kryptografischen Moduls wird mit hoher Wahrscheinlichkeit entdeckt, woraufhin sofort sämtliche kritische Sicherheitsparameter in Klartext mit Nullen überschrieben werden. Kryptografische Module des Sicherheits-Level 4 sind nützlich für den Einsatz in Umgebungen, die anderweitig nicht physisch geschützt sind.

Werfen Sie für weitere Informationen über diese Sicherheits-Level sowie weitere Spezifikationen des FIPS-Standards einen Blick auf den vollständigen FIPS 140-2 Standard unter http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

7.3. National Industrial Security Program Operating Manual (NISPOM)

Das NISPOM (auch DoD 5220.22-M genannt), als Teil des National Industrial Security Program (NISP), legt eine Reihe von Verfahren und Anforderungen für alle Subunternehmer der Regierung bezüglich sensibler Informationen fest. Das aktuelle NISPOM ist vom 28. Februar 2006. Das NISPOM-Dokument kann von der folgenden URL heruntergeladen werden: https://www.dss.mil/GW/ShowBinary/DSS/isp/fac_clear/download_nispom.html.

7.4. Payment Card Industry Data Security Standard (PCI DSS)

Auszug von https://www.pcisecuritystandards.org/about/index.shtml: *Der PCI Security Standards Council ist ein offenes, globales Forum, gegründet in 2006, das sich der Entwicklung, Verwaltung, Unterrichtung und Sensibilisierung für die PCI-Sicherheitsstandards, einschließlich dem Data Security Standard (DSS), verschrieben hat.*

Sie können den PCI DSS Standard unter https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml herunterladen.

7.5. Handbuch zur technischen Sicherheitsimplementierung

Ein Handbuch zur technischen Sicherheitsimplementierung (engl. Security Technical Implementation Guide oder kurz STIG) ist eine Methode zur standardisierten, sicheren Installation und Verwaltung von Computer-Software und -Hardware.

Siehe folgende URL für eine Liste verfügbarer Handbücher: http://iase.disa.mil/stigs/stig/index.html.

Kapitel 8. Weitere Informationsquellen

Die folgenden Links verweisen auf weitere Informationsquellen, die für SELinux und Red Hat Enterprise Linux relevant sind, jedoch über den Rahmen dieses Handbuchs hinausgehen. Beachten Sie, dass aufgrund der schnellen Entwicklung von SELinux einige dieser Materialien nur auf bestimmte Releases von Red Hat Enterprise Linux anwendbar sind.

Bücher

SELinux by Example

Mayer, MacMillan, and Caplan

Prentice Hall, 2007

Tutorials und Hilfen

Verstehen und Anpassen der Apache HTTP SELinux-Richtlinie

http://docs.fedoraproject.org/selinux-apache-fc3/

Tutorials und Vorträge von Russell Coker

http://www.coker.com.au/selinux/talks/ibmtu-2004/

Allgemeines HOWTO zum Schreiben von SELinux-Richtlinien

http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

Red Hat Wissensdatenbank

http://kbase.redhat.com/

Allgemeine Informationen

NSA SELinux Hauptwebsite

http://www.nsa.gov/selinux/

NSA SELinux FAQ

http://www.nsa.gov/selinux/info/faq.cfm

Fedora SELinux FAQ

http://docs.fedoraproject.org/selinux-faq/

SELinux NSA's Open Source Security Enhanced Linux

http://www.oreilly.com/catalog/selinux/

Technologie

Ein Überblick über Objektklassen und Berechtigungen

http://www.tresys.com/selinux/obj_perms_help.html

Integrieren flexibler Unterstützung für Sicherheitsrichtlinien in das Linux-Betriebssystem (eine Geschichte der Flask-Implementierung in Linux)
http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

Implementieren von SELinux als Linux-Sicherheitsmodul
http://www.nsa.gov/research/_files/publications/implementing_selinux.pdf

Eine Sicherheitsrichtlinien-Konfiguration für Security-Enhanced Linux
http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

Community

Fedora SELinux-Benutzerhandbuch
http://docs.fedoraproject.org/en-US/Fedora/13/html/Security-Enhanced_Linux/

Fedora SELinux-Handbuch zur Verwaltung eingeschränkter Dienste
http://docs.fedoraproject.org/en-US/Fedora/13/html/Managing_Confined_Services/

SELinux Community-Seite
http://selinuxproject.org/

IRC
irc.freenode.net, #selinux, #fedora-selinux, #security

Geschichte

Geschichtliches zu Flask
http://www.cs.utah.edu/flux/fluke/html/flask.html

Umfassende Hintergrundinformationen zu Fluke
http://www.cs.utah.edu/flux/fluke/html/index.html

Verschlüsselungsstandards

A.1. Symmetrische Verschlüsselung

A.1.1. Advanced Encryption Standard - AES

In der Kryptografie ist der Advanced Encryption Standard (AES) ein Verschlüsselungsstandard, der von der U.S. Regierung übernommen wurde. Der Standard besteht aus drei Blockchiffren, AES-128, AES-192 und AES-256, die von einer größeren, ursprünglich als Rijndael veröffentlichten Gruppe übernommen wurden. Jede AES-Chiffre hat eine 128-bit Blockgröße, mit Schlüsselgrößen von 128, 192 bzw. 256 Bits. Wie auch schon der Vorgängerstandard Data Encryption Standard (DES) wurden die AES-Chiffren eingehend analysiert und werden nun weltweit eingesetzt. [14]

A.1.1.1. Anwendungsfälle für AES

A.1.1.2. Geschichte von AES

AES wurde am 26. November 2001 vom National Institute of Standards and Technology (NIST) als U.S. FIPS PUB 197 (FIPS 197) angekündigt, nach einem fünfjährigen Standardisierungsprozess, in dessen Verlauf 15 konkurrierende Entwürfe vorgestellt und untersucht wurden, bevor schließlich Rijndael als zweckmäßigster Entwurf ausgewählt wurde (siehe Advanced Encryption Standard Prozess für weitere Einzelheiten). Dieser Standard trat am 26. Mai 2002 in Kraft. Er steht in vielen verschiedenen Verschlüsselungspaketen zur Verfügung. AES ist die erste öffentlich verfügbare und offene Chiffre, die von der NSA für streng geheime Informationen zugelassen wurde (siehe Sicherheit von AES unten). [15]

Die Rijndael-Chiffre wurde von zwei belgischen Kryptografen, Joan Daemen und Vincent Rijmen, entwickelt und für den AES-Auswahlprozess eingereicht. Rijndael (ausgesprochen [rɛinda:l]) ist ein Kofferwort bestehend aus den Namen der beiden Erfinder. [16]

A.1.2. Data Encryption Standard - DES

Der Data Encryption Standard (DES) ist eine Blockchiffre (eine Form der Verschlüsselung mit geheimen Schlüsseln), die im Jahr 1976 als offizieller Standard für die US-Regierung ausgewählt und seither international vielfach eingesetzt wird. Er basiert auf einem symmetrischen Schlüsselalgorithmus, der einen 56-bit Schlüssel verwendet. Der Algorithmus war ursprünglich umstritten aufgrund seiner geheimen Design-Elemente, einer relativ kurze Schlüssellänge und Verdächtigungen über eine integrierte Hintertür für die National Security Agency (NSA). DES wurde infolgedessen einer eingehenden, akademischen Prüfung unterzogen, die unser modernes Verständnis von Blockchiffren und der Kryptoanalyse begründete. [17]

A.1.2.1. Anwendungsfälle für DES

A.1.2.2. Geschichte von DES

Heute gilt der DES aufgrund seiner geringen Schlüssellänge für viele Applikationen als nicht mehr sicher genug. Im Januar 1999 brachen distributed.net in Kooperation mit der Electronic Frontier Foundation einen DES-Schlüssel öffentlich in 22 Stunden und 15 Minuten. Es gibt zudem Untersuchungsergebnisse, die einige theoretische Schwächen in der Chiffre aufzeigen, diese sind in der Praxis jedoch kaum auszunutzen. Der Algorithmus in Form von Triple DES gilt als nahezu sicher, obwohl es auch hier theoretische Schwächen gibt. In den letzten Jahren wurde die Chiffre durch den Advanced Encryption Standard (AES) abgelöst. [18]

In einigen Quellen wird zwischen DES als Standard und DES als Algorithmus (auch Data Encryption

Algorithm oder kurz DEA genannt) unterschieden. Im Sprachgebrauch wird "DES" entweder als Abkürzung ausgesprochen (/ˌdiːˌiːˈɛs/) oder als einsilbiges Akronym (/ˈdɛz/). [19]

A.2. Asymmetrische Verschlüsselung

Die asymmetrische Verschlüsselung, die von vielen kryptografischen Algorithmen und Kryptosystemen eingesetzt wird, verwendet asymmetrische Schlüsselalgorithmen anstelle von oder zusätzlich zu symmetrischen Schlüsselalgorithmen. Mithilfe der asymmetrischen Verschlüsselung sind nun viele Methoden zur Absicherung von Kommunikations- oder Authentifizierungsdaten praktikabel, die vorher undenkbar waren. Auf diese Weise ist der vorherige sichere Austausch von einem oder mehreren geheimen Schlüsseln, der bei der Verwendung symmetrischer Schlüsselalgorithmen unabdingbar ist, hier nicht mehr nötig. Sie kann auch zur Erstellung digitaler Signaturen verwendet werden. [20]

Die asymmetrische Verschlüsselung ist eine grundlegende und weltweit verbreitete Technologie, die auch Internet-Standards wie Transport Layer Security (TLS) (Nachfolger von SSL), PGP und GPG zugrunde liegt. [21]

Asymmetrische Kryptosysteme verwenden asymmetrische Schlüsselalgorithmen, bei denen der Schlüssel zur Verschlüsselung einer Nachricht nicht identisch ist mit dem Schlüssel zur Entschlüsselung. Jeder Benutzer verfügt über ein Paar von kryptografischen Schlüsseln — ein öffentlicher Schlüssel und ein privater Schlüssel. Der private Schlüssel wird geheim gehalten, während der öffentliche Schlüssel weit verbreitet werden kann. Nachrichten, die mit dem öffentlichen Schlüssel des Empfängers verschlüsselt wurden, können nur mit dem dazugehörigen privaten Schlüssel wieder entschlüsselt werden. Obwohl die Schlüssel mathematisch miteinander verwandt sind, kann vom öffentlichen Schlüssel unmöglich der private Schlüssel abgeleitet werden. Die Entdeckung solcher Algorithmen revolutionierte die Kryptografie ab Mitte der 70er Jahre. [22]

In Gegensatz dazu verwenden symmetrische Schlüsselalgorithmen, die in der ein oder anderen Form bereits seit einigen Tausend Jahren genutzt werden, einen einzigen geheimen Schlüssel, der vom Sender und Empfänger gemeinsam verwendet wird (darüber hinaus jedoch geheim gehalten werden muss, was für Verwirrung in der Terminologie sorgen kann) zur Verschlüsselung und zur Entschlüsselung. Um ein symmetrisches Verschlüsselungsschema nutzen zu können, müssen der Sender und der Empfänger im Voraus auf sichere Weise einen Schlüssel austauschen. [23]

Da symmetrische Schlüsselalgorithmen fast immer deutlich weniger rechenintensiv sind, ist es üblich, zum Austausch eines Schlüssels einen Algorithmus zu diesem Zweck zu nutzen und die Daten mithilfe dieses Schlüssel und einem symmetrischen Schlüsselalgorithmus zu übertragen. PGP sowie die SSL/TLS-Familie gehen nach diesem Verfahren vor und werden daher oft Hybrid-Kryptosysteme genannt. [24]

A.2.1. Diffie-Hellman

Der Diffie-Hellman-Schlüsselaustausch (D–H) ist ein kryptografisches Protokoll, mit dem zwei Kommunikationspartner, die sich vorher nicht kennen, über einen unsicheren Kommunikationskanal zusammen einen geheimen, gemeinsam genutzten Schlüssel erzeugen können. Dieser Schlüssel wird anschließend verwendet, um Nachrichten mittels einer symmetrischen Chiffre zu verschlüsseln. [25]

A.2.1.1. Geschichte von Diffie-Hellman

Das Schema wurde zum ersten Mal in 1976 von Whitfield Diffie und Martin Hellman veröffentlicht, obwohl später bekannt wurde, dass es unabhängig davon bereits einige Jahre zuvor innerhalb des britischen Government Communications Headquarters (GCHQ) von Malcolm J. Williamson erfunden wurde, jedoch unter Verschluss gehalten wurde. Im Jahre 2002 schlug Hellman vor, den Algorithmus Diffie–Hellman–Merkle Schlüsselaustausch zu nennen, um den Beitrag von Ralph Merkle zur Erfindung der

asymmetrischen Kryptografie zu würdigen (Hellman, 2002). [26]

Obwohl die Diffie–Hellman-Schlüsselvereinbarung selbst ein anonymes (nicht-authentifiziertes) Schlüsselvereinbarungsprotokoll ist, bildet es die Grundlage für eine Vielzahl an authentifizierten Protokollen und wird eingesetzt, um perfekte Weiterleitungssicherheit in den Transport Layer Security Modi EDH bzw. DHE zu bieten. [27]

Das U.S. Patent 4.200.770, mittlerweile abgelaufen, beschreibt den Algorithmus und nennt Hellman, Diffie und Merkle als Erfinder. [28]

A.2.2. RSA

In der Kryptografie ist RSA (was für ihre Erfinder Rivest, Shamir und Adleman steht) ein Algorithmus der asymmetrischen Kryptografie. Es war der erste Algorithmus, der sowohl zur Signierung als auch zur Verschlüsselung eingesetzt werden konnte und stellte so einen Meilenstein in der asymmetrischen Kryptografie dar. RSA wird weit verbreitet im elektronischen Handel eingesetzt und gilt aufgrund seiner ausreichend langen Schlüssel und der Verwendung von aktuellen Implementierungen als sicher.

A.2.3. DSA

DSA (Digital Signature Algorithm) ist ein Standard der US-Regierung für Digitale Signaturen. Der DSA ist ein reines Signatur-Verfahren, es gibt kein verwandtes Verschlüsselungsverfahren. [29]

A.2.4. SSL/TLS

Transport Layer Security (TLS) und sein Vorgänger Secure Sockets Layer (SSL) sind kryptografische Protokolle, die Sicherheit für Datenübertragungen über Netzwerke wie das Internet ermöglichen. TLS und SSL verschlüsseln die Segmente der Netzwerkverbindungen auf der Transportschicht zwischen zwei Endpunkten.

Mehrere Versionen dieser Protokolle werden weitläufig in Anwendungen wie Webbrowsern, E-Mail, Internet-Fax, Instant Messaging und Voice-Over-IP (VoIP) eingesetzt. [30]

A.2.5. Cramer-Shoup Kryptosystem

Das Cramer–Shoup-System ist ein asymmetrischer Verschlüsselungsalgorithmus und war das erste praktikable Verschlüsselungsverfahren, das im Standardmodell (ohne Zufallsorakel) gegen adaptive Chosen-Ciphertext-Angriffe sicher war. Die Sicherheit des Verfahrens beruht auf der Schwierigkeit des Decisional-Diffie-Hellman-Problems. Es wurde in 1998 von Ronald Cramer und Victor Shoup entwickelt als Erweiterung des Elgamal-Kryptosystems. Im Gegensatz zu dem sehr verformbaren Elgamal fügt Cramer–Shoup zusätzliche Elemente hinzu, um die Nicht-Verformbarkeit selbst gegen hartnäckige Angreifer zu erhöhen. Diese Nicht-Verformbarkeit wird mithilfe einer kollisionsresistenten Hashfunktion und zusätzlichen Berechnungen erreicht, die in zweimal so umfangreichen Chiffraten resultieren. [31]

A.2.6. ElGamal-Verschlüsselung

In der Kryptografie ist das ElGamal-Kryptosystem ein asymmetrischer Verschlüsselungsalgorithmus der asymmetrischen Kryptografie, das auf der Diffie-Hellman Schlüsselvereinbarung basiert. Es wurde in 1985 von Taher Elgamal beschrieben. ElGamal-Verschlüsselung wird in der freien GNU Privacy Guard Software, in aktuellen Versionen von PGP und anderen Kryptosystemen verwendet. [32]

---

[14] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[15] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[16] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[17] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[18] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[19] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[20] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[21] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[22] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[23] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[24] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[25] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[26] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[27] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[28] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[29] "DSA." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[30] "TLS/SSl." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Transport_Layer_Security

[31] "Cramer-Shoup cryptosystem." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Cramer–Shoup_cryptosystem

[32] "ElGamal encryption" *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/ElGamal_encryption

Versionsgeschichte

Version 1.5-2.402	Fri Oct 25 2013	Rüdiger Landmann
Rebuild with Publican 4.0.0		
Version 1.5-2	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
Version 1.5-0	**Apr 19 2010**	**Scott Radvan**
Kleinere Fehlerbehebungen, finaler Entwurf für Beta		
Version 1.4.1-0	**Mar 5 2010**	**Scott Radvan**
QE-Prüfung und Aktualisierungen		
Version 1.3-0	**Feb 19 2010**	**Scott Radvan**
Verlegen auf Testbereich bereit für Prüfung.		